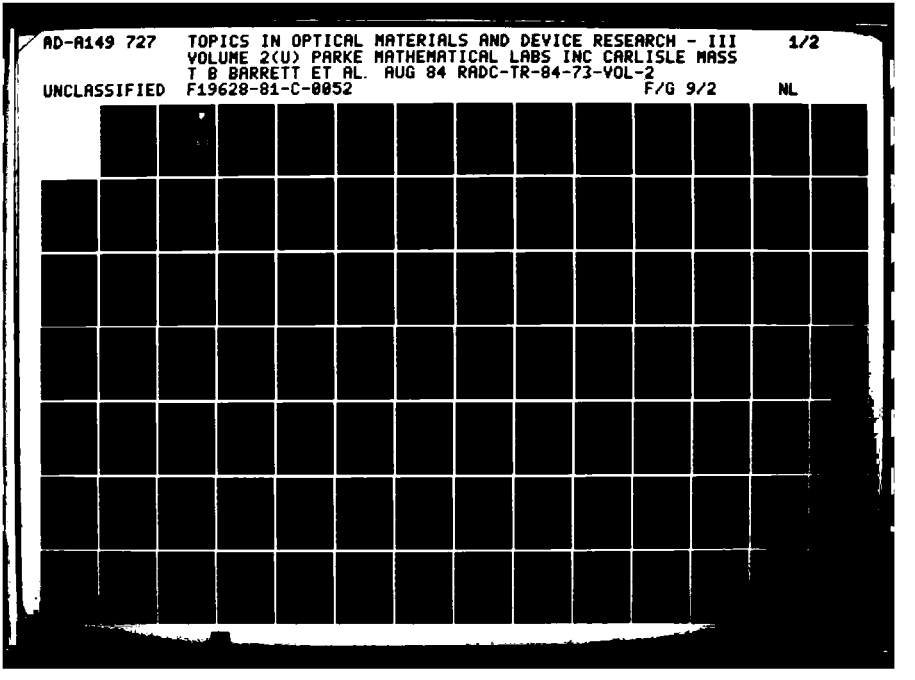
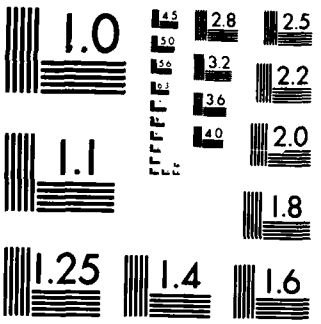

AD-A149 727 TOPICS IN OPTICAL MATERIALS AND DEVICE RESEARCH - III 1/2
VOLUME 2(U) PARKE MATHEMATICAL LABS INC CARLISLE MASS
T B BARRETT ET AL. AUG 84 RADC-TR-84-73-VOL-2
UNCLASSIFIED F19628-81-C-0052 F/G 9/2 NL



(12)

AD-A149 727

RADC-TR-84-73, Vol II (of two)
Final Technical Report
August 1984

TOPICS IN OPTICAL MATERIALS AND DEVICE RESEARCH - III

Parke Mathematical Laboratories, Inc.

T. B. Barrett
R. Marshall
C. Warde
J. Caulfield
M. M. Salour

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-84-73, Volume II (of two) has been reviewed and is approved for publication.

APPROVED: Carl A. Pitha

CARL A. PITHA
Project Engineer

APPROVED: Harold Roth

HAROLD ROTH, Director
Solid State Sciences Division

FOR THE COMMANDER: John A. Ritz

JOHN A. RITZ
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ESO) Hanscom AFB MA 01731. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS N/A
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) N/A	5. MONITORING ORGANIZATION REPORT NUMBER(S) RADC-TR-84-73, Volume II (of two)
6a. NAME OF PERFORMING ORGANIZATION Parke Mathematical Laboratories, Inc. 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Rome Air Development Center (ESO)
6c. ADDRESS (City, State and ZIP Code) 1 River Road Carlisle MA 01741	7b. ADDRESS (City, State and ZIP Code) Hanscom AFB MA 01731
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Rome Air Development Center 8b. OFFICE SYMBOL (If applicable) (ESO)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-81-C-0052
8c. ADDRESS (City, State and ZIP Code) Hanscom AFB MA 01731	10. SOURCE OF FUNDING NOS PROGRAM ELEMENT NO. 62702F; PROJECT NO. 4600; TASK NO 19; WORK UNIT NO 36
11. TITLE (Include Security Classification) TOPICS IN OPTICAL MATERIALS AND DEVICE RESEARCH - III	
12. PERSONAL AUTHOR(S) T. B. Barrett, R. Marshall, C. Warde, J. Caulfield, M. M. Salour	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM Dec 80 TO Dec 83	14. DATE OF REPORT (Yr., Mo., Day) August 1984 15. PAGE COUNT 134
16. SUPPLEMENTARY NOTATION N/A	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	GPIB interface, Nicolet 1080 computer, Hamamatsu camera
20	02, 06		
	12, 14		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Operating instructions, program listings, etc. for a GPIB interface for a Nicolet 1080 computer are given. The design of this interface was given in RADC-TR-81-372, Volume II dated January 1982, (A111481).

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Carl Pitha	22b. TELEPHONE NUMBER (Include Area Code) (617) 861-3458 22c. OFFICE SYMBOL RADC (ESO)

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

FOREWORD

This report is the second volume of a two-volume Final Report for Contract F19628-81-C-0052. Part I is entitled

Topics in Optical Materials and Device Research III.

Table of Contents

Table of Contents (continued)

INTRODUCTION

The purpose of this volume of the Final Report is to update Final Technical Report RADC-TR-81-372 (Vol.II), January 1982 (referred to below as *). This update consists primarily of listings of revised assembly language code for both NIC and CTL; the biggest change being that it is no longer necessary to down load the "operational code" from NIC to CTL, since this code has been stored in the CTL EPROM.

In addition more descriptive material is included on using the NIC software with emphasis on its use to control the Hamamatsu C1000 camera.

As before:

CTL = NIC-488/CTL, a GPIB interface for the Nicolet 1080 series of computers
NIC = the Nicolet 1080 computer
C1000 = the Hamamatsu C1000 camera (or more exactly that camera's controller)

SECTION I Controlling the C1000 using NIC/CTL.

CTL presently contains software in EPROM which allows it to run in "automatic" or "manual" mode. The automatic mode is used mainly to test the camera, bus and CTL - it can not be used at present to usefully gather video data from the C1000. In manual mode, the user has full control of the camera through NIC software which is referred to below as MAIN. As explained in ", MAIN is also used for other functions such as down loading CTL from NIC and for examaning (and changing) code in CTL.

Automatic Mode

To use the automatic mode, the CTL address switch 6 (see appendix A) must be placed in the "bit set" position while either address switch 5 or 4 or both must also be "set" (the other 2 address switch positions are immaterial) the sixth switch 1, numbered, is the SYC switch which should always be "set".

If switch 5 is set, the camera cursor first moves from its default x-coordenate of 512 to 0 and is then stepped to x=1023 and thence to 0.

If switch 4 is set, a complete "frame" of video is gathered using the default interface value of 2 (512 pixels/scan). The frame starts at x=0 if switch 5 is set and at 512 otherwise.

The automatic mode may start when CTL is first turned on (assuming that the C1000 controller is on). Usually, however, CTL will power on in a "frozen" state where nothing can happen before a soft reset is done. (This happens because the various "latches" in CTL go to random states on power on). Thus to reliably use the auromatic mode a reset command should be sent from NIC to CTL. This can be done, for example, by running MAIN in "mode 1" (see next section) or by executing the CTLRS instruction (octal 4072).

Manual Mode

This is the normal mode of operation for camera control. This mode is possible whenever CTL address switch 6 is "off".

Program MAIN should be loaded into NIC by typing the console command RUN NIC CTL. MAIN will issue the query "MODE?-". There are three "modes" having to do with camera control, modes 1,4 and 5.

mode 1

This mode is used to set the system including camera, camera controller, CTL and MAIN in a known "initial" state. In particular the camera parameters are givin their default values. (see table 1)

Previsiosly (see *), mode 1 was also used to down-load the CTL operational software from NIC. Now, however, this software is resident in CTL EPROM. It is still possible to down load software in mode 2, however.

Mode 1 should always be invoked whenever the system is first powered on and whenever a system "hang" condition is encountered.

mode 4

Mode 4 is used to control the camera format (output/input), various camera parameters (x-coordinate, interface, marker on/off) and buffer display. (Horizontal resolution and external anolog select are not available on the present camera). The mode query for mode 4 is "MNEMONIC?-" to which the user may answer with one of the 3 letter mnemonic codes given in table 1. The current numerical value associated with the mnemonic (if any) is displayed to which the user may "reply" with CR if no change in value is to be made, or with the appropriate new numerical value. Note that BUF (buffer display) requires no data and immediately activates the buffer display if BUF is entered.

MNEMONIC	MEANING	DEFAULT VALUE	PERMISSIBLE VALUES	MEANING	NOTE
OUT	Output Format	1	1,2	1=ASCII data 2=binary data	1
INF	Input Format	1	1,2,3	1=ASCII 2=binary 3="print"	2
XCO	X-Coordinate	512	0-1023		3
INT	Interlace	2	1,2,4	1=256 pixel 2=512 pixel 4=1024 pixel	4
MAR	Marker on/off	1	0,1	0=on 1=off	5
BUF	Buffer Display				6

1. The output format is the format (ASCII or binary) of data passed from CTL to the camera. The default value of 1 should not be changed, since MAIN sends all data to the camera in ASCII.

2. The input format is the format of (video) data passed from the camera to CTL. ASCII data means that each 8 bit video value (datum) is sent as 3 ASCII numbers, most significant number first, with ASCII space following each number. Binary data means that each datum is sent as a single 8 bit binary integer. "Print" is the same as ASCII except that a carriage return, line feed is sent in place of every 16th space. INF (3) should never be used since there is not enough buffer space in CTL to receive data in this mode. INF (1) should be used only if INT (1) is also used. (This means that the CTL buffer will be completely filled with 1024 bytes (256 pixels at 4 bytes per pixel.) INF (2) must be used if data is to be stored on MAC disk. On the other hand, if a scan line is to be received on the MAC oscilloscope (or plotted) INF (1) must be used.

3. This is the x-coordinate of the (vertical) video scan line which is digitized by the camera. (If the marker is on, it will be positioned at the current x-coordinate.) Normally, XCO should be set to zero before a frame of video is gathered. It has been found by experiment that the x-coordinate does not change properly if an n-digit (n=2,3,4) number was previously entered and then an m-digit number (m<n) is entered. For example, if a value of XCO (1000) is entered followed by XCO (200) then the final position will be wrong. This quirk of the camera can be overcome by always entering 4 digits of x-coordinate values (e.g. 0010 is permissible for XCO (10)).

Table 1 - A summary of mode and camera "commands".

4. The interlace value refers to the multiplier of 256; thus the default value of 2 means that each video scan line consists of 512 pixels.

5. The marker is a vertical cursor which appears on a camera video monitor to indicate the relative location of the vertical scan line which is digitized.

6. The buffer display is a vertical plot (observable on a camera video monitor) of video intensity vs. y-position for the video scan line currently being digitized. The buffer display appears whenever the camera is reset (using mode 1, for instance) and disappears whenever a command (other than BUF) is issued to the camera.

Table 1- (continued)

mode 5

Mode 5 is used to gather a single line of video or a "frame" of video after the camera has been properly "formatted" using mode 4. The video may be displayed or stored (in packed form) for later manipulation.

The first mode query is "FILE NAME?-" to which the user should reply with an up-to-six character file name (not starting with "PRI") if the video data is to be stored or with "PRINT" (can be abbreviated to "PRI") if the data is to be displayed.

The second mode query is "FRAME?-" to which the user should reply with VI1, VII or VID for 1 line of video, 1 "frame" of video - positive increment, 1 "frame" of video - negative increment. For VI1, a single video scan line is obtained, at the x-coordinate value chosen under mode 3 (or default 512). For VII, a frame of data is gathered, starting at the current x-coordinate and ending at x = 1023. Similaraly for VID except the ending x-coordinate is 0. In both VII and VID the size of the step between increments is controlled by the value of INT; thus if INT=4 the step size is 1, for INT=2 it is 2 and for INT=1 it is 4.

Two forms of "print" (if FILE NAME=PRINT) are used depending on the value of INF. If INF=1 then 16 numerical values per line are printed - each value is separated by a space. If INF=2, 32 hexadecimal values (00 to FF) per line are printed with no spaces. It should be noted that if INF=1, a print file is always chosen, regardless of the value given to FILE NAME. e.i. ASCII data is never filed. Also, if the value of INT and INF is such that CTL buffer overflow will occur (see mode 4) a message saying "CHANGE INT OR INF" is printed.

For details on the form of the packed video file, see Appendix B.

The status byte received from CTL during serial poll is printed each time a video scan line is to be received. In "PRINT" mode this byte is printed on a line above the video data; in "file" mode, a maximum of 36 status bytes per line are printed.

SECTION II - Peripheral Commands for NIC/CTL

In the previous section, modes 1,4 and 5 were described for use in controlling the C1000. In this section, the peripheral "commands" for use with CTL are described. These commands are invoked as modes 2,3,6,7 and 8.

mode 2

This mode may be used to automatically down load Z-80 code (command 8) from NIC to CTL and to cause this code to be executed. However, MAIN itself does not at present include code to execute the "command 8" corresponding to CTL command 8. The two mode queries are "COMMAND#?-" and "FILE NAME?-". The first query is a formality since only 8 can be entered. (Previously two commands were possible.) The file name is the name of the NIC file containing the code to be down loaded to CTL, (see note below) This code may be most conveniently developed on a Tektronix 8002 Development system and then stored on the Comm-Stor as a WHEX file. (see Mode 3) This WHEX file is then transferred to a NIC file using Mode 3.

The code is loaded starting at location 0900H in CTL and should have entry point 0900H. A total of 512 bytes are loaded.

The only other restriction on the code is that is should return via a jump to subroutine RETURN. (use JP 14EH)

The NIC subrountine corresponding to CTL "COM 8" must include a command table transfer followed by a call to MONITOR near the begining and at least one call to MONITOR at the end of the command. (Other calls to MONITOR may be required if data is to be transferred from CTL to NIC.) See NIC "COM 1" or "COM 2" for examples of programs which merely pass the command table to CTL.

Mode 2 is normally used as a temporary expedient to try out new code in the CTL.

NOTE Mode 8 may be used to easily modify Z-80 code before it is transferred to CTL. The code to be modified must be loaded into NIC "data" memory starting at octal 100000. The modified code can then be transferred to CTL by using Mode 2 and replying to "FILE NAME?-" with a non-existent file name.

mode 3

This mode is used to transfer and store Z-80 code from another device to NIC. This code can later be down loaded into CTL using mode 2, for example. The transfer is via the NIC serial input (RS-232) and specifically requires that the code be transferred in the form of a Tektronix WHEX (hexadecimal) file (see *, Appendix D). Mode 3 was

specifically designed to be used with a SYKES Comm-Stor (communication storage unit with floppy disk storage). It can be used however with any computer (or possibly intelligent terminal) which can output the required WHEX file and respond (start sending) to the command "IB file" where file is the user supplied answer to the query "FILE NAME?-". It should be noted that NIC requires 8 bit ASCII with the eighth bit always 1 at various baud rates from 110 to 1200. No other protocal is required.

mode 6

This mode is used to transfer part of CTL memory (in blocks of 256 bytes) to NIC where it can be examined using mode 7 or examined and changed using mode 8.

The mode query is "HEXN-" to which the user should reply with the 4 digit hexadecimal starting address of CTL memory to be examined followed by a single digit representing the number of 256 byte blocks to be transferred.

mode 7

Displays NIC data mamory (starting at 100000) in hexadecimal from (5 nibbles/NIC word) at 32 characters/line. The number of bytes (2 nibbles/byte) displayed is given by the contents of location NBY5 (see symbol table for SEG1A). If mode 3 or 6 is run prior to mode 7, NBY5 contains the number of bytes transferred during mode 3 or 6 operation. (Note that during mode 3 the CTL code is stored in NIC data memory prior to being filed.)

mode 8

Examine and change "hexadecimal" data in NIC memory. This mode may be used to modify the CTL program to be stored in CTL in mode 2 operation. Note that CTL memory locations 0000-07FF (2048 bytes) are in EPROM and can not be changed. (See Appendix C for the map of memory in both NIC and CTL.)

The mode query is OCT-, to which the user should reply with the (up to 7 digit) octal starting address of NIC memory to be examined and possibly changed. (The address given is right justified zero-fill. Thus, 100=0000100. Usually the address will be 100000 plus.)

The contents of memory at this location is then displayed as 5 hexadecimal digits (5 nibbles in each word). The user may respond with

(a) space (actually any sequence of 1 to 4 characters) followed by CR.

(b) 5 characters followed by CR.

(c) CR with no preceding character.

If (a), then the next word is displayed with no change to the current word.

If (b), the current word is changed to the value given and the next word is displayed. (If the value is not hexadecimal, then the changed word is nonsense.)

If (c), then the mode is exited with no change to the current word.

SECTION III Miscellaneous Operating Notes.

There are various "escapes" in MAIN which may be used to change the flow of program execution. These are listed and discussed below.

CONTROL-Q: If control-Q is typed at any time that MAIN is waiting for input from the user e.g. while waiting for a reply to a mode query, a jump is made to NICBUG. The user can then examine memory or jump to the NIC operating system (monitor) by typing 7600G. A return to MAIN can be made from NICBUG by typing 71G.

CONTROL-G: Similar to control-Q except a jump is made to the start of MAIN which will cause MODE?- to be typed. This is a convenient excape from a mis-typed input (Note that rubout does not work in MAIN.)

CONTROL-Z: Must be typed to excape from the "no find" condition in mode 3. The end-of-file character which mode 3 expects is control-Z. If this character is never sent (e.g. if the file does not exist in the sending device), it must be provided by the user.

Any character: If a "hung" condition is encountered, it is usually possible to escape by typing any character. This condition usually occurs either when NIC is trying to communicate with CTL or CTL is trying to communicate with the bus (e.g., the C1000). In the former case, a jump to NICBUG is made; in the latter case, the state of CTL is printed out (see Table 2) and a jump is made to NICBUG. In either case the user should try to determine the cause of the hung condition before continuing. Frequently, all that is necessary is to return to MAIN (use 71G) and use mode 1 to reset.

ROW 1 (CTL Command Table)

Identifier Code	=	Value Name
0	=	COMM
1	=	NLIST
2	=	LIS1P
3	=	LIS1S
4	=	LIS2P
5	=	LIS2S
6	=	LIS3P
7	=	LIS3S
8	=	TALKP
9	=	TALKS
A	=	EOSC
B	=	NDAT
C	=	NDATB
D	=	DATADD (LO)
E	=	DATADD (HI)
F	=	DUM1L
0	=	DUM1H
1	=	MESS
2	=	STAT1
3	=	STAT2
4	=	RETADD (LO)
5	=	RETADD (HI)
6	=	LCOMM
7	=	MAXBLK
8	=	STACKP (LO)
9	=	STACKP (HI)
A	=	COUNT (LO)
B	=	COUNT (HI)

ROW 2 (CTL Registers)

Identifier Code	=	Value Name
L	=	L REGISTER
H	=	H REGISTER
PC	=	PROGRAM COUNTER
SP	=	STACK POINTER
A	=	A REGISTER
C	=	C REGISTER
B	=	B REGISTER
E	=	E REGISTER
D	=	D REGISTER
0	=	8291 INT1
1	=	8291 INT2
2	=	8291 ADRST
3	=	8291 ADR01
4	=	8291 EOSR
5	=	8292 INTST
6	=	8292 ERROR FLAG
7	=	8292 CONTROLLER STATUS
8	=	8292 BUS STATUS

Table 2 - CTL State Data Returned on Interrupt

SECTION IV CTL Software

The original CTL software as outlined in * and listed there has been modified. The major modification is the inclusion of the ability to interrupt program execution and relocation of the program to origin 0. This allows the program to be put in EPROM so it does not have to be down loaded from NIC. Other changes were made. For example, no check is made of the SYC switch to determine if CTL is the controller in charge of the bus.

The CTL software (now actually "firmware") is refered to below as CTLF1 (CTL firmware 1). It consists of the following components:

(1) An initialization routine which sets various registers in the 8291-set and causes the IFC (interface clear) message to be sent on the bus. (Assuming SYC is set.) The routine is always run whenever a software reset is done (CTLRS) and on power on (provided hardware latches permit it).

(2) An "automatic" test module which is invoked if address switch 0 is "on". This module is invoked under the same conditions as the initialization routine.

(3) An interrupt driven routine which transfers CTL state data to locations starting at the end of the command table. (The command table presently starts at the bottom of RAM at 800H.) This routine starts at 38H. Interrupt mode 1 is enabled so that whenever an interrupt occurs, this routine is activated. Note that a NIC CTLWR instruction will cause a Z-80 interrupt if interrupts are enabled. Interrupts are disabled whenever data is being transferred from NIC to CTL.

(4) 6 "command" routines. These command routines are labeled COM 1, COM 2, COM 4, COM 5, COM 6, and COM 7. A seventh "command" (COM 3) may be loaded and run at 900H.

These commands are table driven, i.e. a 20 byte table containing the number of the command to be executed and command parameters is transferred from NIC to CTL prior to command execution. Three of the commands (COM 1, COM 2, COM 4) transfer data to or from the bus while two of the commands (COM 5, COM 6) transfer data to or from NIC. COM 7 is used to transfer data directly from the bus to NIC. A brief description of these "commands" and the required parameters follow:

Bus Commands

These commands always exit with the ATN message inactive so that as long as a bus command is not being executed, the bus is in an "idle" state.

COM 1 (NL1ST, LIS1P, LIS1S, LIS2P, LIS2S, LIS3P, LIS3S, NDAT, NDATB, DATADD)

This command causes data stored in the CTL "buffer" starting at location DATADD (2 bytes) to be transmitted on the bus to NLIST listeners (up to 3) identified by their primary/secondary address (LIS1P/LIS1S, etc.). The number of bytes transferred is determined by NDAT and/or NDATB as follows:

(a) if NDAT and NDATB are both zero no data is sent. Listener addresses are sent however provided NLIST ≠0.

(b) if NDAT ≠ 0, NDAT bytes are sent (0 < NDAT < 255)

(c) if NDAT = 0 and NDATB ≠ 0 then NDATB 256-byte blocks are sent where 0 < NDATB ≤ 4 since up to 1024 bytes may be stored in the CTL buffer. The last byte transferred is accompanied by the END message. (EOI active.)

The NLIST device addresses (pairs) are sent prior to sending the data. If a secondary address is 0, it is not sent.

The universal unlisten message (UNL) is sent after all the data has been transmitted.

COM 2 (TALKP, TALKS, EOSC, NDAT, NDATB, DATADD)

This command transfers data from the bus to the CTL "buffer" starting a DATADD. The amount of data transferred depends on the values of EOSC, NDAT and NDATb. The rules are as follows:

(a) If an END is received with the n^{th} byte and n is less than or equal to the amount of data "expected" (see below), then n bytes are transferred.

(b) Similarly, if EOSC is not zero and a character is received which matches EOSC, n bytes are transferred.

(c) The "expected" amount of data is given by NDAT and NDATB as for COM 1 i.e. if NDAT ≠ 0, then the expected number is NDAT while if NDAT = 0 and NDATB ≠ 0, NDATB 256 byte blocks of data are expected.

(d) If END is not received (or EOSC ≠0 and no match character is received), the "expected" amount of data is transferred.

(e) If NDAT = NDATB = 0 the "expected" amount of data is considered to be unknown. In this case the number of bytes transferred is given by the above rules except that NDATB is assigned an implicit value of 4 (the maximum number of blocks

which can be contained in the CTL buffer). In addition, if END (or matching EOSC) is recevied, the amount of data received is "returned" in NDATB and the first byte of DATADD: (NDATB contains the least significant byte of the 2 byte value returned).

Before the data is gathered from the bus, the talker address given by TALKP/TALKS is transmitted to the bus. If TALKS = 0, no secondary talk address is transmitted.

After data has been gathered, the universal untalk message (UNT) is transmitted to the bus.

COM 4 (LIS1P, LIS1S)

This command responds to a request for service, polls (serial poll) the device with address LIS1P/LIS1S (LIS1S = 0 means no secondary address) and returns the serial poll "status byte" to NIC. (The "status byte" consists of the uniline message, RQS, on data line 7 and STB on data lines 1-6 and 8. If the RQS message is true, the addressed device is requesting service; otherwise it is not. Status information may or may not be passed in the multi-line STB message. The C1000 passes no status information.)

It is noteworthy that the C1000 never requests service if ATN is inactive. Thus the first action in COM 4 is to activate ATN using the TCNTR command to the 8292. (This may cause problems with other devices.)

After the SRQ message is received by COM 4, the serial poll enable, SPE, message is sent followed by the address of the device to be polled. Next CTL goes to stand by to receive the "status byte" and return it to NIC provided CTL is not in the "automatic" (test) mode.

Next the universal untalk message (UNT) is sent followed by the serial poll disable message (SPD).

COM 7 (TALKP, TALKS)

This command passes each datum from the bus directly to NIC without using the CTL buffer. Thus, in a sense it is a combination of COM 2 and COM 6. Like COMM 2 it addresses the talker using TALKP and TALKS. Using COM 7 it is possible to receive "files" of data longer than 1024 bytes.

COM 7 ends reception when the END message is received (EOI is asserted) with the last datum. When this happens the SRVC bit in the CTRLO port is set such that the NIC instruction CTLRD of CTLRDC will "see" the 8th bit of ACC set to 1. It is reset before COM 7 returns.

Non-bus Commands

COM 5 (NDAT, NDATB, DATADD)

This command transfers data stored in CTL starting at DATADD to NIC. The use of NDAT and NDATB are as for COM 1.

COM 6 (NDAT, NDATB, DATADD)

This is the inverse of COM 5 i.e., data is transferred from NIC to CTL "buffer".

SECTION V Software Maintenance

In this section some miscellaneous information concerning software maintenance for both CTL and NIC is presented.

NIC Software

All NIC software was written in Nicolet Assembly Language using the Disk Editor (DISKED), and assembled using the Nicolet Disk Assembler (ASM), (see Appendix E). The source program MAIN is written in 6 segments which should be individually assembled. Any given segment is completely within a page (pages are 1024 words long starting at address 0), but there may be more than one segment per page (see Appendix C). Entry points external to a given segment are always given at the end of the segment (in a list of externals) where they are relatively easily accessible for change. It is important to make note of values assigned to these entry points so that software changes in one segment causing a change of entry point address will be changed in the list of externals of other segments.

The first segment (now called SEG1A) but latter versions may have a different suffix letter) is partitioned into 3 sub-segments called SEG1A1, SEG1A2, SEG1A3 for ease of transfer to backup disks. (A maximum of 4 complete tracks can be stored in data memory.) Before assembly these sub-segments should be combined into one segment using the DISKED K command.

The source software is also stored on floppy disk (in the Sykes Comm-Stor) using the DISKED L command on the complete source program. (All segments are combined using DISKED and the command K, NICP, SEG1A, SEG2A, SEG3, SEG4, SEG5, SEG6.)

Assuming that NICP has been created and DISKED has been invoked, the file can be transferred using the Comm-Stor command !RØNICP#LØNICP.

! = Comm-Stor prompt.

R = Read and store from modem port.

NICP = Name to be assigned to the file (the name does not have to be the same as that being transferred.)

= Separator for a command to be sent out of the modem port prior to data reception.

L NICP is the DISKED command "list file NICP".

In order to transfer this program back to NIC, a NIC program

called COMNIC is available (see Appendix D). If COMNIC is not available in "run" form it should be assembled using ASM and overlayed on program MAIN since COMNIC uses several subroutines in MAIN. Assume MAIN has been loaded, simply use RUN LOADER followed by COMNIC:G.

COMNIC transmits a send command to Comm-Stor in the form !SØfile-name where file-name is given by the user in response to the query "FILE NAME?-. If the file exists in Comm-Stor, it will be transferred to NIC and stored under file-name:A. This new file will then be essentially identical to the one originally transferred from NIC and may be edited using DISKED.

CTL Software

The CTL software was written on a Tektronix 8002A Micro-Processor Lab using the resident editor with Zilog mnemonics and assembled using the resident assembler (see Appendix F). The source program is stored in 6 segments, partly for logical reasons and partly so that any given segment will fit into the 300 line buffer of the editor. The segments are as follows:

CTLSYM1 - The list of equates.

TTLMAC1 - Macro definitions.

CTLMAIN1 - Main control routine and ancillary routines.

TTLSUBS1 - All command subroutines except COM1.

TTLSUB01 - Command subroutine COM1 and the automatic TEST routine along with its BCD to ASCII routine.

CTLSTOR - Parameter table and other RAM storage definitions.

These source subroutines are stored on floppy disk with back up disks.

The machine code can be transferred to CTL in two ways. One method is via Tektronix WHEX files as described under mode 3 in Section II. This method is used for code that is to be down loaded into CTL RAM. The other method is to transfer the code into EPROM as described below.

The PROM burner used at PML is on a CP/M based system. Thus it is necessary to transfer machine code from the 8002A to a file in the CP/M based system. This is accomplished using the 8002A COMM command (see Tektronix 8001/8002 System User's Newsletter - Issue #11, Spring 1980) to send the file and program TEKREC (listed in Appendix G) to receive the file. Modem port (J101) is used with

COMM - see figure 1 for the appropriate cabling diagram.

PIN#	8002 FUNCTION	CODE		CP/M FUNCTION	CODE
2	Transmitted Data	TXD	——▷——	Received Data	RXD
3	Received Data	RXD	——◁——	Transmitted Data	TXD
5	Clear to send *	CTS	——◁——	Request to Send	RTS
7	Signal ground	GND	——▷◁——	Signal Ground	GND
8	Signal Detect	DCD	——◁——	Data Terminal Ready	DTR

* A jumper on the Communication board may be changed to use PIN 6.

Notes

(1) CTS must be on before the start of data transmission (before COMM is called). It may be set off during transmission to halt transmission.

(2) DCD must be on before and during data transmission.

(3) PIN 4 is the 8002 RTS line which is always ON. It may be used by the CP/M system if necessary.

Figure 1 - RS-232 Cabling Diagram for communications with the Tektronix 8002A J101 Modem Port.

The file transfer is initiated by first loading the assembled code to be transferred using the command LOAD "file", where "file" is the name of the file to be loaded. (Optionally, the memory area to be loaded can first be filled with hex FF so that areas of memory which will not contain code will contain FF. Use FILL start-address end-address FF.)

Next use the COMM command:

COMM P=0A.

This is followed with a request for a Formatted Upload by typing:

NULL start-address end-address CR

(NULL is SHIFT/CONTROL @)

APPENDIX A - Address Switch Description

The address switch chip contains six single pole, double throw switches. These switches are numbered (left to right) from 1 to 6. (Note switch 1 corresponds to "logical" switch 0.)

Switch 1 is the SYC switch which may be used to initialize CTL as the system controller in charge of the bus. Normally (with the present software) this switch should be in the "set" position or "off" as marked on the switch. (Set means that SYC is active.)

Switches 2-6 can be used to set the primary address of the CTL, switch 6 corresponds to the least significant bit of the 5 bit address etc. A switch in the off position means that the corresponding bit is set (to 1).

With the current software, the CTL can not be addressed as a talker/listener since it controls what devices are to talk and/or listen - CTL knows it is a listener if another device is asked to talk. Thus the address switches are not used as such, but are instead used to determine "automatic" or "manual" mode of operation of CTL as described in Section I. Again, a bit is "set" by putting the corresponding switch in the "off" position.

APPENDIX B Video Data Files

Each pixel in a video frame (one or more scan lines of video) is an 8 bit byte representing a relative intensily value from 0 to 255. These bytes are packed 2.5 bytes per NIC 20 bit word. NIC data memory (starting at 100000) is used as a buffer to hold 4 tracks worth of data before it is written to the disk. These 4 tracks hold 6144 words or 15360 bytes of video data. This corresponds to 15, 30 and 60 scan lines at INT values 4, 2, 1 respectively. When unpacking the data for display (or other) purposes it is necessary to know the number of scan lines, n, and the value of INT which was used. This infromation (and more) is containing in the file directory as follows:

Directory Word	Interpretation
Core buffer address	Starting x-coordinate (binary form)
File size (words)	File size (words), (note 1,2)
Program starting address (PSA)	n and INT, (note 3)

The file starting track number is, of course, also included in the directory.

Note 1 - The last track will seldom be completely filled with video data.

Note 2 - The file size is actually the number of tracks times 1536 (3000 octal).

Note 3 - The interlace number (1, 2 or 4) is stored in bits 11-14 of the PSA and the number of lines, n, in bits 0-10.

APPENDIX C Memory Maps

	NIC Octal Addresses	Memory Map
	0	
	⋮	Command Table
	23	
	71	
	⋮	Camera "Main"
SEG1A	1432	
	1500	
	⋮	Command Subroutines
	1664	
	1665	
	⋮	Externals, Definitions and Common
	1740	

	2020 : 2343	Error and other Subroutines
SEG2A		
	2344 : 2367	Externals, Definitions and Command
	2420 : 2554	Disk I/o Routines
SEG3		
	2555 : 2567	Externals and Switch Storage
	2605 : 3303	Packing Routines, etc.
SEG4		
	3304	External Reference
	3310 : 3744	Tekhex "Routines"
SEG5		
	3745 : 3760	Externals, definition and scratch storage

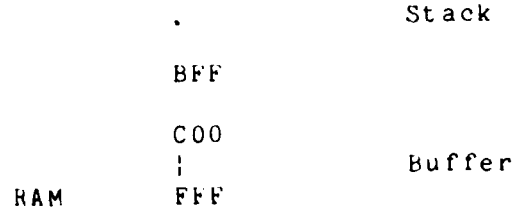
	4010 ¦	Miscellaneous Routines
SEG6	4411	
	4412 ¦	Externals and Definitions
	4425	
	4632 ¦	Nicbug II
	5365	
	5366 ¦	Not used
	5777	

6000 ¦ 7577	Not used
7600 ¦ 7777	Monitor Head

Locations 100000 - 117777 are used for data storage. Note that the Nicolet computer has a 70000 (octal) gap between "program memory" and "data memory".

CTL Memory Map

	Hex Addresses	
	0	
	¦	Intialization
	24	
	38	
	¦	Interrupt Routine
CTLMAIN1	7F	
	80	
	¦	Command Calls, etc.
	BF	
	C0	
	¦	Miscellaneous Routines
	16F	
	174	
TTLSUBS1	¦	All Command Subroutines Except COM1
	323	
	324	
TTLSUB01	¦	COM1 and TEST
	41C	
	41D	
	¦	Unused
ROM	7FF	
	800	
	¦	Command Table
	813	
	814	
	¦	Miscellaneous Data
	81B	
	81C	
	¦	Temporary State Storage on Interrupt
	82F	
	.	
	.	


Stack
BFF
C00
Buffer
RAM
FFF

APPENDIX D - COMNIC Source Listing

```
/       PROGRAM COMNIC                                  ( OMNIC
/ REVISION  -  JUNE 8,1981
/ AUTHOR    -  BARRETT,TB                               ( NIC -80)
/ PURPOSE   -  READ A DISKED A-FILE FROM COMM STOR
/       TO NIC AND PACK IT FOR USE BY DISKED.
/       NOTE    THE COMM-STOR FILE IS PRODUCED BY USING THE "L"
/       COMMAND IN DISKED.
/ QUERIES   - FILE NAME?
/       REPLY WITH THE NAME OF THE COMM STOR FILE TO BE TRANSFERRED.
/       THE PACKED FILE WILL BE GIVEN THE SAME NAME.
/ CAUTION     THE FILE IS STORED AFTER ALL OTHER FILES ON THE DISK.
/       NO CHECK IS MADE FOR ADEQUATE SPACE.

/       COMM STOR MUST BE CONFIGURED TO RESPOND TO  S FOR STOP
/       TRANSMISSION (#19) AND TO  Q FOR START (#20).
/       THE FILE MUST USE  Z FOR EOF (#8) AND COMM-STOR MUST
/       SEND IT (#9).

/ NOTE THE SECTION OF CODE STARTING WITH CHKCH. THIS CODE
/ IS USED TO REPLACE ALL CONTROL CHARACTERS EXCEPT CR,TAB,LF WITH  Z
/ IF THIS IS NOT DONE, DISKED WILL MISS ENTIRE BLOCKS OF
/ INPUT TEXT.
/ ... / ... ////////////// ////////////// ////////////////////////////
        DATAD-POINTER+1
COMNIC, JMS     @CRLF
        JMS     @ONF
        14
        0
        TEXT    "% FILE NAME?=%
        JMS     @PFB
        FIL1
NW,     0
        JMS     @CRLF
        MEMA    (3
        XFMA    NW
        ACLM    NW1
        ZERM    @DATA0.
        JMS     @DEFENW
        MEMA    @DATA0.1
        XCLM    1100
        ACLM    11
        MONM    STATE   /INITIALIZATION
        MEMA    (1
        ACLM    COUNT1
        ZERM    EOF
        ZERM    NBYTT
        JMS     ZRMEM
        JMS     @ONF    /SEND 'S FILE-NAME
NW1,    0
        1
        16300
FIL1,   0
FIL2,   0
        0270000
COM2,   JMS     @PAFF
        COMSTO
NBY,    100000
        0
        100000
NBYTT,  0
        JMS     @WRITE
11,     0
INDEX,  5000
DOBS,   100000
```

```
          MEMA    IT
          MEMA    NBYTO
          A+MM    NBYTI
          MEMZ    EOF       / IF EOF SET,WRAP UP.
          JMP     COM3
          JMS     ZRMEM
          MEMA    (221      / 0 TO RESTART COMMISTOR
          JMS     @TYPE
          JMP     COM2
COM3,     MEMA    ITO
          ACCM    @OARG1
          MEMA    IT
          A+MA    ITO
          JMS     @MULTF
          2000
          ACCM    @OARG2
          MEMA    BUFS
          ACCM    @OARG3
          MEMA    FIL1
          ACCM    FILN1
          MEMA    FIL2
          A+MA    EXTA      /EXTENSION A
          ACCM    FILN2
          JMS     @CLOSE
FILN1,    0
FILN2,    0
          JMS     @EXEC
          JMS     @UNF
          25
          0
          TEXT    % NO. OF PACKED BYTES=%
          MEMA    NBYTI
          JMS     @PRTDEC
          JMP     @SYSTEM

/ END OF MAIN PROGRAM

ZRMEM,    0
          MEMA    BUFS
          ACCM    POINT
          MEMA    1BUFSZ
          ACCM    COUNT
COM1,     ZERM    @POINT
          MFCM    POINT
          MMCMZ   COUNT
          JMP     COM1
          JMP     @ZRMEM

COMSTO,   0
          MEMA    STATE
          AMCZ              / IF STATE =0,   DON'T GET DATA
          ZERZ
          JMP     S3
          AFCIA   /CHECK FOR S4 STATE FOR "ESCAPE" CHARS.
          AFCIAZ
          ZERZ
          JMP     S4
          TTYRF
          JMP     # 1
          RDTTY
          A-MZ    (221      / 0
          ZERZ
          JMP     @SYSTEM   /THIS IS A USER ESCAPE
          A-MZ    (2        / 2 MEANS EOF
          JMP     S12       /CHECK FOR S1 OR S2 STATE
          ONEM    EOF
```

```
        DNEM    @RECOUNT
        MEMA    (-14
        JMP     COMSTE
51.     MOVZ    STATE   /S1 OR S2?
        JMP     S2
        MMOMZ   COUNT1  /S1 STATE
        JMP     CHRCK
        MEOM    STATE
        JMP     CHRCK
S2.     A-MZ    (215
        JMP     CHRCK
        MEMA    (217    / S TO STOP COMM-STOR
        JMS     @TYPE
        JMS     MOCHR
        MEMA    (215
        MEOM    STATE
        DNEM    @RECOUNT
        MEOM    @RECOUNT
        JMP     COMSTE
S3.     MONM    STATE   /S3 STATE
        MEMZ    NCH
        MMOM    STATE   /SET S4 STATE
        MEMA    (1
        A-MA    NCH
        ACCM    COUNT1
        MEMA    (-14
        JMP     COMSTE
S4.     MMOMZ   NCH
        ZERZ
        MONM    STATE
        MEMA    @POINT1
        MEOM    POINT1
CHRCK.  A-MZ    (211
        ZERZ
        JMP     COMSTE
        A-MZ    (212
        ZERZ
        JMP     COMSTE
        A-MZ    (215
        ZERZ
        JMP     COMSTE
        ACCM    TEMP
        A-MA    (240
        EXCT    +0019
        JMP     BADC
        MEMA    TEMP
        JMP     COMSTE
BADC.   MEMA    (240
COMSTE. JMP     @COMSTO

MOCHR.  0
        MEMA    C1      /USE C1 AS TIME OUT CONSTANT
        +ACCM   TEMP
        ZERM    NCH
        MEMA    (DATAD
        +ACCM   POINT1
MOCHR1. TTYRF
        JMP     MOCHR2
        RDTTY
        +ACCM   @POINT1
        MEOM    POINT1
        MEOM    NCH
        JMP     MOCHR1
MOCHR2. MMOMZ   TEMP    DECREMENT TIME OUT COUNTER
        JMP     MOCHR1
        MEMA    (DATAD  /RESTORE THE POINTER FOR GETTING CODE CHARS.
```

```
        ACCM    POINT1
        JMP     @MUCHR
/
/ SCRATCH STORAGE
IIO,    0
POINT,  0
COUNT,  0
COUNT1, 0
NBYTT,  0
STATE,  0
EOF,    0
TEMP,   0
NCH,    0
POINT1, 0
        0
        0
        0
        0
        0
        0
        0
        0
/
/       EXTERNAL REFERENCES
CNTF,   2736
UNF,    2650
FER,    3240
OPENW,  2420
PARF,   3074
TYPE,   2731
WRITE,  2470
MULTF,  4175
CLOSE,  2451
PRTOCT, 2605
OARG1,  7770
OARG2,  7771
OARG3,  7772
SYSTRT, 7600
PCOUNT, 3223
/
/CONSTANTS
EXTA,   1000000
C1,     6700
$
```

APPENDIX E - MAIN Source Listing

```
#
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
//  CAMERA MAIN (NOVEMBER 6,1981)
//  MAIN DRIVER PROGRAM FOR CAMERA CONTROL. THIS PROGRAM PERFORMS
/ VARIOUS BASIC OPERATIONS HAVING TO DO WITH CONTROLLING THE
/ HAMAMATSU C1000 CAMERA VIA THE NIC-488/CTL INTERFACE.
/ IN ADDITION VARIOUS CONTROLLER FUNCTIONS ARE PERFORMED. EACH FUNCTION
/ IS CALLED A MODE (OF OPERATION). AT PRESENT 7 MODES HAVE BEEN
/ IMPLEMENTED AS FOLLOWS:
/
/ MODE1 - RESET THE CAMERA,CONTROLLER (NO LONGER HAVE TO BOOT LOAD)
/ THIS OPERATION SHOULD USUALLY BE THE FIRST OPERATION ON SYSTEM
/ START AND MAY BE REPEATED ANY TIME THE USER WISHES TO
/ IN RAM WHICH MUST BE RELOADED EACH TIME THE CTL IS TURNED OFF.)
/
/ MODE2 - LOAD AND EXECUTE SPECIAL "COMMANDS". AT PRESENT THE
/ CTL RECOGNIZES ONE OPTIONAL COMMAND (NO. 8). THIS
/ COMMAND IS NOT PART OF THE MAIN SEQUENCE OF CTL PROGRAMS AND
/ MUST BE LOADED BEFORE CALLING. AT PRESENT, IT IS ASSUMED THAT THE
/ CORRESPONDING NIC COMMAND PROGRAM IS IN NIC CORE. THE MODE QUERY IS-
/       COMMAND #?- (REPLY  8)
/       FILE NAME?- (REPLY WITH THE NIC FILE TO BE LOADED)
/       [OPTIONAL SPECIAL DATA, DEPENDING ON THE COMMAND]
/
/ MODE3 - TRANSFER AND TRANSFORM A TEKTRONIX 8002 TEK-HEX
/ FILE FROM COMM-STOR TO A NIC LOAD FILE. THIS MODE PROVIDES FOR
/ EASY TRANSFER OF ASSEMBLED Z-80 CODE TO NIC AND THENCE TO CTL.
/ THE MODE QUERY IS -
/       FILE NAME?- (RESPOND WITH THE COMM-STOR FILE NAME WHICH
/                ALSO BECOMES THE NIC FILE NAME)
/ NOTE- IF THE FILE DOES NOT EXIST IN COMMSTOR, THE USER WILL
/ BE INFORMED OF THIS. HE MUST THEN TYPE ^Z TO GET PROGRAM CONTROL.
/
/ MODE4 - CAMERA SET-UP. USING THIS MODE, ANY OF 5 LEGITIMATE
/ CAMERA SET-UP COMMANDS GIVEN IN CAMERA TABLE ARE SENT TO THE CAMERA.
/ THE MODE QUERY IS -
/       MNEMONIC?- (REPLY WITH ONE OF THE 3 LETTER MNEMONIC
/       CODES: OUT INF XCO INT MAR
/ THE CURRENT VALUE IS THEN PRINTED AFTER WHICH THE USER MAY TYPE
/ IN A NEW VALUE OR CR TO RETAIN THE GIVEN ONE.
/ NOTE THAT AT PRESENT THE USER SHOULD NOT CHANGE OUT WHICH GIVES
/ THE FORMAT OF DATA SENT TO THE CAMERA. IT SHOULD STAY AT
/ ITS DEFAULT VALUE OF 1 MEANING ASCII DATA.
/
/ MODE5 - OBTAIN A "FRAME" OF VIDEO DATA (THE SIZE OF A FRAME DEPENDS
/ ON THE CAMERA TABLE VALUES FOR XCO AND INT AND ON THE COMMAND).
/ THE MODE QUERY IS -
```

```
/       FILE NAME?- (REPLY WITH THE NAME OF THE NIC FILE TO RECEIVE
/       THE DATA OR WITH "PRINT" WHICH WILL CAUSE THE DATA TO BE
/       PRINTED BUT NOT STORED. IF "PRINT" IS GIVEN, THE ACTUAL FORM
/       OF THE PRINTED VIDEO DEPENDS ON THE INPUT FORMAT (INF). IF 1 OR 3,
/       THE DISPLAY IS 3 CHARACTER (000-256) FOLLOWED BY "BLANK", 16
/       PIXELS/LINE. IF 2, THE DISPLAY IS 2 CHARACTER HEX (00-FF) WITH
/       NO BLANKS AND 32 PIXELS/LINE.)
/       FRAME?- (REPLY WITH VI1,VII,VID FOR 1-LINE VIDEO,A FRAME
/       STARTING AT X-COORD GOING TO HIGHER X-VALUES, A FRAME
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
/       STARTING AT X-COORD GOING TO LOWER X-VALUES RESPECTIVELY.
/
/ MODE6 - TRANSFER CTL MEMORY TO NIC AT 100000 IN PACKED FORM.
/ THE MODE QUERY IS -
/       HEXN- (REPLY WITH 4 HEX DIGIT ADDRESS AND A SINGLE DIGIT
/              SPECIFYING THE NO. OF 256 BYTE BLOCKS WANTED.
/            E.G. 0C004 MEANS ADDRESS C00 AND 4 BLOCKS)
/ THE DATA OBTAINED MAY BE DISPLAYED BY CALLING MODE7
/
/MODE7 - DISPLAY MEMORY IN HEX FORMAT. THE REGION DDISPLAYED STARTS
/ AT 100000 AND IS NW3 WORDS LONG WHERE NW3 IS OBTAINED BY RUNNING MODE3
/ OR MODE6. THE USER CAN ALSO CHANGE NW3 HOWEVER.
/ THE WORDS ARE ASSUMED TO BE IN PACKED FORM.
/
/MODE8 - DISPLAY NIC WORDS IN HEX AND PERMIT THE USER TO CHANGE THE
/ DISPLAYED WORD (ALSO IN HEX). THIS MODE CAN BE USED TO PUT PROGRAM
 / PATCHES INTO CTL.
/ THE MODE QUERY IS -
/       OCT - ( REPLY WITH THE STARTING ADDRESS (OCTAL) OF NIC
/            MEMORY TO BE OBSERVED AND/OR CHANGED )
/       THE PROGRAM THEN DISPLAYS THE FIRST WORD AS 5 HEX DIGITS AND
/       WAITS FOR USER RESPONSE. A SPACE (OR ANY SEQUENCE OF 1 TO 4
/       CHARACTERS) THEN CR WILL CAUSE NO CHANGE.
/       IF 5 HEX CHARACTERS ARE ENTERED THEY WILL REPLACE THE DISPLAYED
/       WORD. THE NEXT WORD WILL THEN BE DISPLAYED, ETC. TO EXIT THIS
/       MODE REPLY WITH CR ONLY. THE CHANGED CODE CAN BE
/       LOADED INTO CTL MEMORY VIA MODE 1 WITH A NON-EXISTANT FILE
/       NAME.
/
/
/ THE USER MAY EXIT CTLSYS VIA NICBUG TO NICSYS AND STORE THIS PROGRAM
/ DATA FOR LATER USE. USE STORE NAME 100000-100632;100000:P
/
/ NOTE -- WHEN RESPONDING TO A QUERY, THE USER MAY TYPE ^G TO ABORT
/ THE MODE. TYPING ^Q WILL BRING THE USER TO NICBUG (IF LOADED).
/ VALUE ASSIGNED TO ABORT.
// PARAMETER TABLE
      0       0 COMN,    0
```

```
      1         0 NLIST,   0
      2         0 LIS1P,   0
      3         0 LIS1S,   0
      4         0 LIS2P,   0
      5         0 LIS2S,   0
      6         0 LIS3P,   0
      7         0 LIS3S,   0
     10         0 TALKP,   0
     11         0 TALKS,   0
     12         0 EOSC,    0
     13         0 NDAT,    0
     14         0 NDATB,   0
     15         0 DATAL,   0
     16         0 DATAH,   0
     17         0 RETAL,   0
     20         0 RETAH,   0
     21         0 MESS,    0
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
     22         0 DUM1,    0
     23         0 DUM2,    0
/
/
/        DEFINITIONS
         LISTEN=40          /OCTAL BASE FOR LISTENERS
         TALK=100           /OCTAL BASE FOR TALK
         CTL=1    /ADDRESS ASSIGNED TO CONTROLLER
         CAMERA=2           /ADDRESS ASSIGNED TO CAMERA
         BASE=140           /BASE FOR CAMERA "SECONDARY ADDRESSES"
         OUTF=1   /OUTPUT FORMAT (1,2)
         INF=2    /INPUT FORMAT (1,2,3)
         XCOORD=3           /X-COORDINATE (0 - 1023)
         INTERL=4           /INTERLACE (1,2,4)
         HORRES=5           /HORIZONTAL RESOLUTION (1,2,3,4)
         EXTAN=6 /EXTERNAL ANALOG (1=OFF)
         MARK=7   /MARKER ON/OFF (1=ON)
         VIDIN=10           /VIDEO INPUT
         VIDINI=11          /VIDEO IN & INCREMENT
         VIDIND=12          /VIDEO IN &DECREMENT
         SLICE=13           /SLICE INPUT
         SLICEI=14          /SLICE IN & INCREMENT
         SLICED=15          /SLICE IN & DECREMENT
         BUFFER=16          /BUFFER DISPLAY/
/
         CTLCF=4062
         CTLRD=44064
         CTLRDC=44066
         CTLSK=6064
         CTLWR=4071
```

```
        CTLRS=4072

// CAMERA TABLE (DEFINES CAMERA COMMANDS AND MNEMONICS)
/ EACH ENTRY CONTAINS THE FOLLOWING DATA IN THE SEQUENCE SHOWN-
/       MNEMONIC - 3 LETTERS IN PACKED ASCII (RIGHT JUSTIFIED)
/       CODE - OCTAL SECONDARY ADDRESS DEFINING THE COMMAND
/       NNIB - NO. OF NIBBLES IN THE COMMAND DATA (0-5)
/       C-DATA - UP TO 5 NIBBLES OF NUMERICAL DATA (LEFT JUST.)
     24  576564 CTABLE, 0576564 /OUTPUT FORMAT (OUT)
     25     141          BASE+OUTF
     26       1          1
     27  200000          0200000 /DEFAULT 1
     30  515646          0515646 /INPUT FORMAT (INF)
     31     142          BASE+INF
     32       1          1
     33  200000          0200000 /DEFAULT 1
     34  704357          0704357 /X-COORD (XCO)
     35     143          BASE+XCOORD
     36       3          3
     37 1211000          1211000 /DEFAULT 512
     40  515664          0515664 /INTERLACE (INT)
     41     144          BASE+INTERL
     42       1          1
     43  400000          0400000 /DEFAULT 2
     44  554162          0554162 /MARKER ON/OFF (MAR)
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
     45     147          BASE+MARK
     46       1          1
     47       0          0       /DEFAULT 0
     50  665121          0665121 /1-LINE VIDEO (VI1)
     51     150          BASE+VIDIN
     52       0          0
     53       0          0
     54  665151          0665151 /1-LINE VIDEO + INCREMENT (VII)
     55     151          BASE+VIDINI
     56       0          0
     57       0          0
     60  665144          0665144 /1-LINE+DECREMENT (VID)
     61     152          BASE+VIDIND
     62       0          0
     63       0          0
     64  426546          0426546 /BUFFER DISPLAY (BUF)
     65     156          BASE+BUFFER
     66       0          0
     67       0          0
     70     300          300     /0 TO TERMINATE THE TABLE
     71 3001700 START,   JMS     @CRLF
     72 3001676          JMS     @UNP
```

```
     73        7           7
     74        0           0
     75     5557           TEXT % MO
     76   444537  DE?
     77   157700  -%
    100  3001667 RPT,     JMS       @ECHO
    101   462223           A-MZ      (223      /CONTROL G
    102  2162000           ZERZ
    103  1001730           JMP       @ABORT
    104   462261           A-MZ      ("1
    105  2162000           ZERZ
    106      134           JMP       MODE1
    107   462262           A-MZ      ("2
    110  2162000           ZERZ
    111      216           JMP       MODE2
    112   462263           A-MZ      ("3
    113  2162000           ZERZ
    114      263           JMP       MODE3
    115   462264           A-MZ      ("4
    116  2162000           ZERZ
    117      365           JMP       MODE4
    120   462265           A-MZ      ("5
    121  2162000           ZERZ
    122      560           JMP       MODE5
    123   462266           A-MZ      ("6
    124  2162000           ZERZ
    125     1265           JMP       MODE6
    126   462267           A-MZ      ("7
    127  2162000           ZERZ
    130     1325           JMP       MODE7
    131   462270           A-MZ      ("8
    132       71           JMP       START
    133     1345           JMP       MODE8
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
/
/
////////////////////////////////////////////////////////////////////////
    134  2170000 MODE1,   ZERA      /DUMMY
    135  2410000           ACCA      /IF NO FILE, ASSUME IT IS ALREADY IN CORE (I
    136  3001711           JMS       @ZERTAB /ZERO THE COMMAND TABLE
    137        0           TABLA
/ INITIALIZE THE CAMERA TABLE TO DEFAULT VALUES
    140  3001710           JMS       @SEARCH
    141   576564           0576564
    142        0           0
    143   200000           0200000
    144        1           1
    145       24           CTABLE
```

```
146         1           1
147 3001710             JMS       @SEARCH
150  515646             0515646
151       0             0
152  200000             0200000
153       1             1
154      24             CTABLE
155       1             1
156 3001710             JMS       @SEARCH
157  704357             0704357
160       0             0
161 1211000             1211000
162       3             3
163      24             CTABLE
164       1             1
165 3001710             JMS       @SEARCH
166  515664             0515664
167       0             0
170  400000             0400000
171       1             1
172      24             CTABLE
173       1             1
174 3001710             JMS       @SEARCH
175  554162             0554162
176       0             0
177       0             0
200       1             1
201      24             CTABLE
202       1             1
203    4072             CTLRS     /RESET THE CTL
204 2170000             ZERA      /FORCE CTL TO BOOT LOAD
205 2165663             ZERM      SCTLC
206 2170000             ZERA
207 2170000             ZERA
210 2170000             ZERA
211 2170000             ZERA
212 2170000             ZERA
213 2170000             ZERA
214    4062             CTLCF     /CLEAR
215      71             JMP       START

--------
/ CAMERA MAIN (NOVEMBER 6,1981)
/
////////////////////////////////////////////////////////////////
/
    216 3001715 MODE2,          JMS       @GETFIL /PUT FILE IN NIC CORE
    217      71                 JMP       START   /ERROR EXIT FROM GETFIL
    220 3001700                 JMS       @CRLF
```

```
    221 3001676 MODE21, JMS     @UNP
    222      13         13
    223       0         0
    224    4357         TEXT % CO
    225  555541  MMA
    226  564403  ND#
    227  371577  ?-%
    230 3001711         JMS     @ZERTAB
    231       0         TABLA
    232 3001667         JMS     @ECHO
    233 2405413         ACCM    TEMP     /COMMAND SHOULD BE ASCII 270
    234  462270         A-MZ    (270
    235     240         JMP     #+3
    236  110011         MEMA    (11      /STORE AT 900H
    237     243         JMP     #+4
    240  462270         A-MZ    (270
    241     221         JMP     MODE21   /WRONG COMMAND #
    242  110011         MEMA    (11      /STORE AT 900H
    243 2404016         ACCM    DATAH
    244 2111724         MEMA    BUFS     /SET UP COMMAND TABLE FOR FILE TRANS
    245 2404001         ACCM    NLIST
    246  110005         MEMA    (5
    247 2404000         ACCM    COMN
    250  110002         MEMA    (2       /ASSUME 2 256 BYTE BLOCKS FOR TRANSF
    251 2404014         ACCM    NDATB
    252 2164022         ZERM    DUM1
    253 2001541         JMS     COM5
    254 2111413         MEMA    TEMP
    255  462270         A-MZ    (270
    256     261         JMP     #+3
NL  COM8    AT      257
    257 3000257         JMS     @COM8
    260     262         JMP     #+2
NL  COM8    AT      261
    261 3000257         JMS     @COM8
    262      71         JMP     START
 /
/
//////////////////////////////////////////////////////////////
/
    263 2000352 MODE3,  JMS     FILEQ
    264 3001703         JMS     @PKR
    265     277         FIL1
    266       0 NW1,    0
    267 3001700         JMS     @CRLF
    270  110003         MEMA    (3
    271 2510266         A+MA    NW1
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
```

```
 272 2404274            ACCM    NW2
 273 3001676            JMS     @UNP    /SEND COMMAND TO COMM-STOR TO SEND I
 274       0 NW2,       0
 275       1            1
 276   16300            16300
 277       0 FIL1,      BLOCK   2
 301  770000            0770000
 302 3001702            JMS     @PAKE   /RECEIVE THE DATA AND PACK IT
 303     342            COMSTO
 304  100000            100000
 305       0            0
 306  100000            100000
 307       0 NBYTES,    0
 310 2112307            MEMAZ   NBYTES
 311 2404316            ACCM    NW3
 312  470004            A-MA    (4
 313    5144            EXCT    AC19
 314     263            JMP     MODE3
 315 3001706            JMS     @TEKHEX
 316       0 NW3,       0
 317       0 NBYS,      0
 320 2110316            MEMA    NW3
 321 2404325            ACCM    NW4
 322 2111724            MEMA    BUFS
 323 3405722            ACCM    @OARG3
 324 3001707            JMS     @NICFIL
 325       0 NW4,       0
 326     277            FIL1
 327     331            JMP     MODE31  /ERROR RETURN FROM NICFIL-NO ROOM
 330      71            JMP     START
 331 3001676 MODE31,    JMS     @UNP
 332      21            21
 333    5657            TEXT % NO
 334    6257   RO
 335  575500   OM
 336  465762   FOR
 337    4651   FI
 340  544577   LE%
 341 1001730            JMP     @ABORT
 342       0 COMSTO,    0
 343    6454            TTYRF
 344     343            JMP     .-1
 345   44453            RDTTY
 346  462232            A-MZ    (232    /CHECK FOR ^Z (END OF FILE)
 347 1000342            JMP     @COMSTO
 350 3025732            ONEM    @PCOUNT
 351 1000342            JMP     @COMSTO
/ NOTE THAT SETTING PCOUNT TO 1 FORCES PAKE TO STOP. COMM-STOR
/               SHOULD BE CONFIGURED TO SEND EOF.
```

```
// SUBROUTINE FILEQ
/ PURPOSE -- SEND MESSAGE - FILE NAME?-
    352        0 FILEQ,   0
    353  3001700          JMS      @CRLF
    354  3001676          JMS      @UNP
    355       14          14
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
    356        0          0
    357     4651          TEXT     % FI
    360   544500  LE
    361   564155  NAM
    362   453715  E?-
    363   770000  %
    364  1000352          JMP      @FILEQ
/
/
////////////////////////////////////////////////////////////////////
/
    365  3001711 MODE4,   JMS      @ZERTAB
    366        0          TABLA
    367   110005          MEMA     (5
    370  2404000          ACCM     COMN      /INITIALIZE COMMAND TABLE FOR DATA
    371   110311          MEMA     (311      /FROM NIC TO CTL
    372  2404001          ACCM     NLIST
    373   110014          MEMA     (14
    374  2404016          ACCM     DATAH
    375  2144022          MONM     DUM1      /WILL BE SENDING MESSAGES
    376  3001676 MODE41,  JMS      @UNP
    377       13          13
    400        0          0
    401     5556          TEXT % MN
    402   455557  EMO
    403   565143  NIC
    404   371577  ?-%
    405  3001703          JMS      @PKR
    406      416          FIL2
    407        0 NC,      0
    410  3001700          JMS      @CRLF
    411  2164423          ZERM     FLAG
    412  2110407          MEMA     NC
    413   462003          A-MZ     (3        /MUST HAVE 3 CHARS.
    414      376          JMP      MODE41
    415  3001710          JMS      @SEARCH
    416        0 FIL2,    0
    417        0 CODE,    0
    420        0 VAL,     0
    421        0 NBYTE,   0
    422       24          CTABLE
```

```
     423        0 FLAG,    0
     424  2102423          MEMZ      FLAG
     425      376          JMP       MODE41   /NON-ZERO FLAG MEANS MNEMONIC NOT F
     426  2112421          MEMAZ     NBYTE
     427      431          JMP       #+2
     430      475          JMP       MODE42
     431  2404436          ACCM      NBY1
     432  2110416          MEMA      FIL2
     433  2404466          ACCM      MM1
     434  3001701          JMS       @UNPF    /DISPLAYS THE NIBBLES
     435     2330          HEXTA     /TYPE1 CONVERTS 4 BIT HEX TO NIC-ASCII AND
     436        0 NBY1,    0
     437  3777777          7777777 /-1 FOR NIBBLES
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
     440      420          VAL
     441   110240          MEMA      (240
     442  3001677          JMS       @TYPE
     443  3001677          JMS       @TYPE
     444  3001677          JMS       @TYPE
     445  2000505          JMS       NIB      /PACK THE USER GIVEN NIBBLES
     446      470          VAL4
     447        0 NBY2,    0
     450  3001700          JMS       @CRLF
     451  2112447          MEMAZ     NBY2     /IF NO BYTES, USE DEFAULT
     452      460          JMP       #+6
     453  2110420          MEMA      VAL
     454  2404002          ACCM      LIS1P
     455  2110421          MEMA      NBYTE
     456  2404013          ACCM      NDAT
     457      474          JMP       MODE43
     460  2404013          ACCM      NDAT
     461  2404471          ACCM      NBY4
     462  2110470          MEMA      VAL4
     463  2404002          ACCM      LIS1P
     464  2024473          ONEM      FLG4
     465  3001710          JMS       @SEARCH /UPDATE THE C-TABLE WITH THE NEW VAL
     466        0 MM1,     0
     467        0          0
     470        0 VAL4,    0
     471        0 NBY4,    0
     472       24          CTABLE
     473        0 FLG4,    0
     474  2001541 MODE43,  JMS       COMS
     475  2024000 MODE42,  ONEM      COMN     /SET UP COMMAND TABLE FOR CTL-CAM
     476  2024001          ONEM      NLIST
     477   110042          MEMA      (LISTEN+CAMERA
     500  2404002          ACCM      LIS1P
     501  2110417          MEMA      CODE
```

```
     502 2404003           ACCM     LIS1S
     503 2001500           JMS      COM1
     504       71 MODE44,  JMP      START
////////////////////////////////////////////////////////
// SUBROUTINE NIB(VALA,NNIB)
/ PURPOSE -- PACKS NNIB USER GIVEN NUMBERS INTO THE LOCATION
/       GIVEN BY VALUA. UP TO 5 NIBBLES ,LEFT JUSTIFIED 0 FILL
/       MAY BE PACKED. THE NUMBER OF NIBBLES IS RETURNED IN NNIB
/       AND IS DETERMINED BY CR.
     505         0 NIB,    0
     506 3001702           JMS      @PAKF
     507       552         ECHO1
     510         6         6
     511         0         0
     512      1770         PENDA
     513         0 NNIB,   0
     514 2110513           MEMA     NNIB
     515 3404547           ACCM     @COUNTN
     516  110006           MEMA     (6
     517 3404546           ACCM     @WCNT
     520 2711723           MMOA     PEND
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
     521 3404544           ACCM     @WPNT
     522 3164545           ZERM     @WORD
     523 2111723           MEMA     PEND
     524 3544551           AMOM     @APNT
     525 3024550           ONEM     @BCNT
     526 3001704           JMS      @DEC
     527 3110505           MEMA     @NIB
     530 2405413           ACCM     TEMP
     531 3110545           MEMA     @WORD
     532 3405413           ACCM     @TEMP
     533 2124505           MPOM     NIB
     534 2110513           MEMA     NNIB
     535 3544505           AMOM     @NIB      /-1 BECAUSE OF CR
     536  462006           A-MZ     (6        /IF HAVE FULL WORD, GET FROM PEND
     537     542           JMP      NIB3
     540 3111723           MEMA     @PEND
     541 3405413           ACCM     @TEMP
     542 2124505 NIB3,     MPOM     NIB
     543 1000505           JMP      @NIB
/COMMON DEFINITION
     544    3414 WPNT,     WPNTD
     545    3415 WORD,     WPNTD+1
     546    3416 WCNT,     WPNTD+2
     547    3417 COUNTN,   WPNTD+3
     550    3420 BCNT,     WPNTD+4
     551    3421 APNT,     WPNTD+5
```

```
////////////////////////////////////////////////////////}
    552        0 ECHO1,  0
    553  3001667         JMS     @ECHO
    554   462215         A-MZ    (215
    555  1000552         JMP     @ECHO1
    556  3025732         ONEM    @PCOUNT
    557  1000552         JMP     @ECHO1
 /
/
////////////////////////////////////////////////////////////////
/
/
    560  3001711 MODE5,  JMS     @ZERTAB
    561        0         TABLA
    562  2111734         MEMA    T3       /SET DIVIDE ARGUMENTS
    563  2405202         ACCM    REM2
    564   110005         MEMA    (5
    565  2405005         ACCM    REM
    566  2405170         ACCM    REM1
    567   110044         MEMA    (44      /SET POLL STATUS BYTE COUNTER
    570  2405540         ACCM    NCNT
    571  2000352         JMS     FILEQ    /GET FILE NAME, MAY BE "PRINT"
    572  3001703         JMS     @PKR
    573     1260         FILNM
    574        0         0
    575  3001700         JMS     @CRLF
    576  3001710         JMS     @SEARCH /GET SOME CAMERA PARAMETERS TO DETEF
    577   704357         0704357 /SPACE REQUIREMENTS
    600        0         0
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
    601        0 VALM5,  0
    602        0 NBM5,   0
    603       24         CTABLE
    604        0         0
    605  2110601         MEMA    VALM5
    606  2404612         ACCM    VALM51
    607  2110602         MEMA    NBM5
    610  2404613         ACCM    NBM51
    611  3001705         JMS     @NIBBIN /CONVERT TO BINARY INTEGER
    612        0 VALM51, 0
    613        0 NBM51,  0
    614       12         12
    615        0 BIN,    0
    616  3001710         JMS     @SEARCH /LOOK AT INTERLACE TO GET NO. OF ELE
    617   515664         0515664
    620        0         0
    621        0 VALM52, 0
    622        0         0
```

```
     623       24            CTABLE
     624        0            0
     625  2110621            MEMA     VALM52  /GET INTERLACE NIBBLE AND CONVERT TO
     626     5044            LLSH     4
     627  2404676            ACCM     INT
     630  2025414            ONEM     N       /N IS THE NO. OF LINES/FRAME; DEFAUL
     631  3001676  MOD50,    JMS      @UNP
     632       10            10
     633        0            0
     634     4662            TEXT % FR
     635   415545   AME
     636   371577   ?-%
     637  3001703            JMS      @PKR
     640     1415            FILM5
     641        0            0
     642  3001700            JMS      @CRLF
     643  2111415            MEMA     FILM5
     644  2404647            ACCM     FILM51
     645  2164654            ZERM     FLGM5
     646  3001710            JMS      @SEARCH /GET CORRESPONDING CODE
     647        0  FILM51,    0
     650        0  CODM5,     0
     651        0             0
     652        0             0
     653       24            CTABLE
     654        0  FLGM5,     0
     655  2702654            MMOZ     FLGM5   /FLAG=1 MEANS WRONG COMMAND
     656      660            JMP      #+2
     657      631            JMP      MOD50
     660  2110650            MEMA     CODM5
     661   462150            A-MZ     (BASE+VIDIN
     662      664            JMP      #+2
     663      701            JMP      MODE51  /N=1 FOR VIDIN,ELSE CALCULATE N=
     664   462151            A-MZ     (BASE+VIDINI
     665      672            JMP      MODE52  /(1024-XCOORD)*INT/4 OR
     666   111777            MEMA     (1777   / (XCOORD+1)*INT/4
     667  2430000            APOA
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
     670  2470615            A-MA     BIN
     671      675            JMP      MODE53
     672   462152  MODE52,    A-MZ     (BASE+VIDIND
     673      631            JMP      MOD50
     674  2130615            MPOA     BIN
     675  3001712  MODE53,    JMS      @MULTP
     676        0  INT,       0
     677   405022            RISH     2       /DIVIDE BY 4
     700  2405414            ACCM     N
     701  3001710  MODE51,    JMS      @SEARCH / FIND THE NO. OF BYTES/LINE
```

```
    702  515646          0515646 /INF
    703       0          0
    704       0 VALM53,  0
    705       0          0
    706      24          CTABLE
    707       0          0
    710 2165430          ZERM    FACT1
    711 2110704          MEMA    VALM53
    712    5044          LLSH    4       /CHANGE TO 1,2 OR 3
    713 2404704          ACCM    VALM53
    714  110004          MEMA    (4      /NO. OF BYTES/LINE=256*INT*
    715 2404730          ACCM    FACTOR  / (4 OR 1 OR 4  +  0 OR 0 OR 1/16)
    716 2110704          MEMA    VALM53
    717  462002          A-MZ    (2
    720     722          JMP     #+2
    721 2024730          ONEM    FACTOR
    722  462003          A-MZ    (3
    723     726          JMP     #+3
    724  110020          MEMA    (20
    725 2405430          ACCM    FACT1
    726 2110676          MEMA    INT
    727 3001712          JMS     @MULTP
    730       0 FACTOR,  0
    731 2405416          ACCM    NDATB1  /NO. OF 256 BYTE BLOCKS/LINE
    732 3001712          JMS     @MULTP
    733     400          400
    734 2511430          A+MA    FACT1
    735 2405213          ACCM    NWORD   /TOTAL NO. OF BYTES/LINE
/       CHECK FOR CTL BUFFER OVERFLOW
    736 2111740          MEMA    BFSZ
    737 2471213          A-MA    NWORD
    740    5104          SKIP    AC19
    741     755          JMP     MOD53A
    742 3001676          JMS     @UNP
    743      22          22
    744       1          1
    745    4350          TEXT % CH
    746  415647  ANG
    747  450051  E I
    750  564600  NF
    751  576200  OR
    752  515664  INT
    753  770000  %
    754    1264          JMP     MOD5E
/ IF INF IS NOT 2 OR IF THE FILE NAME IS PRI(NT), THE CAMERA
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
/ VIDEO IS PRINTED ONLY AND NO FILE IS CREATED ON THE DISK.
    755 2025417 MOD53A, ONEM    PRTFLG  /1 MEANS THIS IS A "PRINT FILE"
```

```
     756 3165725           ZERM    @SMODE  /SET SENDF MODE SWITCH
     757  110100           MEMA    (100
     760 3405726           ACCM    @SWCNT
     761 3405727           ACCM    @SWCNTO
     762 2110704           MEMA    VALM53
     763  462002           A-MZ    (2
     764    1045           JMP     MODE59  /IF INF .NE. 2 THEN IS PRINT FILE BY
     765 2111260           MEMA    FILNM   /SEE IF PRINT
     766 2463733           A-MZ    PRI     /PRI=PACKED "PRI"
     767     775           JMP     MOD54
     770 3025725           ONEM    @SMODE
     771  110040           MEMA    (40
     772 3405726           ACCM    @SWCNT
     773 3405727           ACCM    @SWCNTO
     774    1045           JMP     MODE59
/ FOR A DISK FILE , WE HAVE TO ESTIMATE THE SPACE REQUIRED, OPEN THE
/ FILE ETC.
     775 2165417 MOD54,    ZERM    PRTFLG
     776 2111213           MEMA    NWORD   /BYTES/LINE*N*TOTAL NO. OF BYTES
     777 2405003           ACCM    #+4
    1000 2111414           MEMA    N       /THEY ARE REDUCED BY 2/5 IN PACKING
    1001    5001           LASH    1       /MULT BY 2
    1002 3001712           JMS     @MULTP
    1003       0           0
    1004 3001713           JMS     @DIVDE  /DIVIDE BY 5
    1005       0 REM,      0
    1006 2425421           APOM    SIZE    /MUST BE A REMAINDER OF AT LEAST 1
    1007 2111005           MEMA    REM     /TAKE CARE OF THE REMAINDER
    1010  470003           A-MA    (3
    1011    5104           SKIP    AC19
    1012 2125421           MPOM    SIZE
    1013 3165721           ZERM    @OARG2  /DELETE FILE IF IT EXISTS
    1014 2111260           MEMA    FILNM
    1015 2405021           ACCM    FILNM1
    1016 2111261           MEMA    FILNM+1
    1017 2405022           ACCM    FILNM1+1
    1020 3001672           JMS     @CLOSE
    1021       0           FILNM1, 0
    1022       0           0
    1023 3165720           ZERM    @OARG1
    1024 2111421           MEMA    SIZE    /TOTAL NO. OF PACKED WORDS
    1025 3405721           ACCM    @OARG2
    1026 3001670           JMS     @OPENW  /CHECKS TO MAKE SURE THERE IS ROOM
    1027 3111721           MEMA    @OARG2
    1030 2471421           A-MA    SIZE
    1031    5104           SKIP    AC19
    1032    1042           JMP     MODE56
    1033 3001676           JMS     @UNP
    1034      10           10
```

```
   1035       1          1
   1036    5657          TEXT % NO
   1037    6257    RO
   1040  575577   OM%
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
   1041    1264          JMP     MODSE
   1042 3111720 MODE56, MEMA     @OARG1
   1043 2405422          ACCM    ITO       /STARTING TRACK FOR THE FILE
   1044 2405224          ACCM    IT
   1045 2110676 MODE59, MEMA     INT       /GET APPROPRIATE VALUES ASSOCIATED W
   1046  462004          A-MZ    (4        /LIMITS FOR STORING THE DATA
   1047    1056          JMP     MODE57    /AND FOR PAUSE
   1050  110017          MEMA    (17
   1051 2405420          ACCM    J1
   1052  110004          MEMA    (4
   1053 2405423          ACCM    L1
   1054 2111737          MEMA    PA4
   1055    1073          JMP     MODE55
   1056  462002 MODE57, A-MZ     (2
   1057    1066          JMP     MODE58
   1060  110036          MEMA    (36
   1061 2405420          ACCM    J1
   1062  110002          MEMA    (2
   1063 2405423          ACCM    L1
   1064 2111736          MEMA    PA2
   1065    1073          JMP     MODE55
   1066  110074 MODE58, MEMA     (74
   1067 2405420          ACCM    J1
   1070  110020          MEMA    (20
   1071 2405423          ACCM    L1
   1072 2111735          MEMA    PA1
   1073 2405126 MODE55, ACCM     PAUC
   1074 2111414          MEMA    N         /SET COUNTER WITH THE TOTAL NO. OF L
   1075 2405424          ACCM    COUNT
/SET UP THE FIXED PORTION OF THE COMMAND TABLE
   1076  110014          MEMA    (14
   1077 2404016          ACCM    DATAH
   1100  110102          MEMA    (TALK+CAMERA
   1101 2404010          ACCM    TALKP
   1102 2164011          ZERM    TALKS
   1103  110150          MEMA    (BASE+VIDIN      /VI1 FIRST TIME THROUGH
   1104 2404003          ACCM    LIS1S
   1105 2024022          ONEM    DUM1     /DATA IS PACKED UNLESS PRINTED
   1106 2703417          MMOZ    PRTFLG
   1107 2164022          ZERM    DUM1
   1110 2165425 MODE5I, ZERM     NBYTS    /I LOOP
   1111 2111420          MEMA    J1
   1112 2405426          ACCM    J
```

```
   1113 2024000 MODE5J, ONEM     COMN     /SEND
   1114 2024001         ONEM     NLIST
   1115 2164013         ZERM     NDAT
   1116 2164014         ZERM     NDATB
   1117  110042         MEMA     (LISTEN+CAMERA
   1120 2404002         ACCM     LIS1P
   1121 2001500         JMS      COM1
   1122  110004         MEMA     (4       /POLL
   1123 2404000         ACCM     COMN
   1124 2164003         ZERM     LIS1S
   1125 3001716         JMS      @PAUSE
   1126       0 PAUC,   0
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
   1127 2001516         JMS      COM4
   1130  110002         MEMA     (2       /RECV
   1131 2404000         ACCM     COMN
   1132 2111416         MEMA     NDATB1
   1133 2404014         ACCM     NDATB
   1134 2001507         JMS      COM2
   1135  110006         MEMA     (6       /NICO
   1136 2404000         ACCM     COMN
   1137 2111724         MEMA     BUFS
   1140 2404001         ACCM     NLIST
   1141 2111426         MEMA     J
   1142 2463420         A-MZ     J1
   1143 2144001         MONM     NLIST    /DON'T RESET COM6 EXCEPT WHEN J=1
   1144 2001604         JMS      COM6
   1145 2110650         MEMA     CODM5    /FOR NEXT PASSES
   1146 2404003         ACCM     LIS1S
   1147 2111213         MEMA     NWORD
   1150 2505425         A+MM     NBYT5    /ACCUMULATE TOTAL NO. OF BYTES
   1151 2707424         MMOMZ    COUNT
   1152    1154         JMP      #+2
   1153    1164         JMP      MOD5J1
   1154 2707426         MMOMZ    J
   1155 2162000         ZERZ
   1156    1161         JMP      MOD5J0
   1157 2703417         MMOZ     PRTFLG
   1160    1113         JMP      MODE5J
   1161  110004 MOD5J0, MEMA     (4
   1162 2405427         ACCM     NTRCK
   1163    1206         JMP      MOD5J2
   1164 2111425 MOD5J1, MEMA     NBYT5
   1165 3001712         JMS      @MULTP
   1166       2         2
   1167 3001713         JMS      @DIVDE   /DIVIDE BY 5
   1170       0 REM1,   0
   1171 2425431         APOM     SUM      /MUST HAVE A REMAINDER
```

```
1172 2111170          MEMA    REM1
1173  470003          A-MA    (3
1174    5104          SKIP    AC19
1175 2125431          MPOM    SUM
1176 2170000          ZERA    /GET READY FOR DIVIDE
1177    4354          TACMQ
1200 2111431          MEMA    SUM
1201 3001713          JMS     @DIVDE
1202       0 REM2,    0
1203 2405427          ACCM    NTRCK
1204 2103202          MEMZ    REM2
1205 2125427          MPOM    NTRCK
1206 2703417 MOD5J2,  MMOZ    PRTFLG  /PRINT OR STORE?
1207    1217          JMP     MOD5J3
1210 3001700          JMS     @CRLF
1211 3001701          JMS     @UNPF
1212    4242          SENDFA
1213       0 NWORD,   0
1214       1          1       /DATA IS NOT PACKED FOR PRINT
1215  100000          100000
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
1216    1234          JMP     MOD5J4
1217 2111427 MOD5J3,  MEMA    NTRCK   /STORE THE PACKED DATA ON NTRCKS TR. -
1220 2405432          ACCM    K
1221 2111724          MEMA    BUFS    /POINTS TO START OF BUFFER AREA
1222 2405226          ACCM    ISTART
1223 3001673 MOD5K,   JMS     @WRITE  /START OF K LOOP
1224       0 IT,      0
1225    3000 TRKSZ,   3000
1226       0 ISTART,  0
1227 2125224          MPOM    IT
1230 2111225          MEMA    TRKSZ
1231 2505226          A+MM    ISTART
1232 2707432          MMOMZ   K
1233    1223          JMP     MOD5K
1234 2103424 MOD5J4,  MEMZ    COUNT
1235    1110          JMP     MODE5I  /END OF I-LOOP
1236 2703417          MMOZ    PRTFLG  /IF THIS IS A PRINT FLAG GOTO THE E
1237    1241          JMP     #+2
1240    1264          JMP     MOD5E
1241 3111731          MEMA    @SYSTRT /SAVE TO RESTORE
1242 2405413          ACCM    TEMP
1243 2110676          MEMA    INT     /CLOSE THE FILE
1244    5070          RLSH    10      /THE INTERLACE NO. IS STORED AS THE
1245 2511414          A+MA    N       /ORDER BITS OF THE PSA PORTION OF TI
1246 3405731          ACCM    @SYSTRT /DIRECTORY ENTRY; THE NO. OF LINES
1247 2111422          MEMA    ITO     /IN THE LOW ORDER BITS
1250 3405720          ACCM    @OARG1
```

```
   1251 2111224          MEMA    IT
   1252 2471422          A-MA    ITO
   1253 3405722          ACCM    @OARG3  /~~NO. OF TRACKS~~ STORED IN CORE BUFFE
   1254 3001712          JMS     @MULTP
   1255    3000          3000
   1256 3405721          ACCM    @OARG2
   1257 3001672          JMS     @CLOSE
   1260       0 FILNM,   0
   1261       0          0
   1262 2111413          MEMA    TEMP
   1263 3405731          ACCM    @SYSTRT
   1264      71 MODSE,   JMP     START
LIST
/
   1265 3001676 MODE6,   JMS     @UNP
   1266       5          5
   1267       0          0
   1270  504570          TEXT %HEX
   1271  561577  N-%
   1272 2000505          JMS     NIB
   1273    1414          N
   1274       0 NBY6,    0
   1275 2111274          MEMA    NBY6
   1276  462005          A-MZ    (5
   1277    1265          JMP     MODE6   /MUST GET 4 NIBBLES FOR ADDRESS AND
   1300 2111414          MEMA    N       /1 FOR NO. OF BLOCKS (EG. 0C001)
   1301   10007          ANDA    (7
   1302 2404014          ACCM    NDATB
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
   1303 2164013          ZERM    NDAT
   1304 2111414          MEMA    N
   1305    5064          RLSH    4       /GET LOW ADDRESS
   1306   10377          ANDA    (377    /MASK IT
   1307 2404015          ACCM    DATAL   /STORE FOR TRANSMISSION
   1310 2111414          MEMA    N
   1311    5074          RLSH    14      /GET HIGH ADDRESS
   1312   10377          ANDA    (377
   1313 2404016          ACCM    DATAH
   1314 2164022          ZERM    DUM1
   1315 2111724          MEMA    BUFS
   1316 2404001          ACCM    NLIST
   1317  110006          MEMA    (6
   1320 2404000          ACCM    COMN
   1321 2001604          JMS     COM6
   1322 2111640          MEMA    NBYT1   /TRANSFER NO. OF BYTES RECEIVED
   1323 2404317          ACCM    NBY5    /FOR POSSIBLE MODE 7 CALL.
   1324      71          JMP     START
         /
```

```
   1325 3025725 MODE7,  ONEM     @SMODE  /SET UP FOR SENDF
   1326  110020         MEMA     (20
   1327 3405726         ACCM     @SWCNT
   1330 3405727         ACCM     @SWCNTO
   1331 3001700         JMS      @CRLF
   1332 3001700         JMS      @CRLF
   1333 2110317         MEMA     NBY5    /NO. OF BYTES
   1334 2405337         ACCM     MOD71
   1335 3001701         JMS      @UNPF
   1336    4242         SENDFA
   1337       0 MOD71,  0
   1340       0         0
   1341  100000         100000
   1342 3001700         JMS      @CRLF
   1343 3001700         JMS      @CRLF
   1344      71         JMP      START
/
   1345 3001676 MODE8,  JMS      @UNP
   1346       4         4
   1347       0         0
   1350  574364         TEXT     %OCT
   1351  157700  -%
   1352 3001717         JMS      @OCT    /PACK OCTAL ADDRESS INTO MOD8A
   1353    1364         MOD8A
   1354 2711364         MMOA     MOD8A   /STORE POINTER FOR CHANGE WORDS
   1355 2405424         ACCM     COUNT
   1356 2165426         ZERM     J       /FOR COUNTING WORDS
   1357 3001700 MOD81,  JMS      @CRLF
   1360 3001701         JMS      @UNPF   /DISPLAY CONTENTS OF NEXT WORD
   1361    2330         HEXTA            /AS 5 HEX CHARACTERS
   1362       5         5
   1363 3777777         3777777
   1364       0 MOD8A,  0
   1365 2145364         MONM     MOD8A   /CAUSES UNPF TO KEEP GOING WITHOUT
   1366  110240         MEMA     (240    /REINITIALIZING
   1367 3001677         JMS      @TYPE   /PUT IN SPACE
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
   1370 2000505         JMS      NIB     /COLLECT THE NIBBLES
   1371    1414         N
   1372       0 NBY8,   0
   1373 2125424         MPOM     COUNT
   1374 2125426         MPOM     J
   1375 2113372         MEMAZ    NBY8    /0 NIBBLES MEANS EXIT
   1376 2162000         ZERZ
   1377    1405         JMP      MOD8E
   1400  462005         A-MZ     (5      /1-4 NIBS MEANS NO CHANGE
   1401    1357         JMP      MOD81
   1402 2111414         MEMA     N
```

```
   1403 3405424          ACCM    @COUNT
   1404    1357          JMP     MOD81
   1405 2111426 MOD8E,   MEMA    J
   1406 3001712          JMS     @MULTP
   1407       5          5
   1410    5021          RASH    1       /NO. OF BYTES=5/2 * NO. OF WORDS
   1411 2404317          ACCM    NBY5
   1412      71          JMP     START
////////////////////////////////////////////////////////////////////}
/
/SCRATCH STORAGE
   1413       0 TEMP,    0
   1414       0 N,       0
   1415       0 FILM5,   0
   1416       0 NDATB1,  0
   1417       0 PRTFLG,  0
   1420       0 J1,      0
   1421       0 SIZE,    0
   1422       0 IT0,     0
   1423       0 L1,      0
   1424       0 COUNT,   0
   1425       0 NBYT5,   0
   1426       0 J,       0
   1427       0 NTRCK,   0
   1430       0 FACT1,   0
   1431       0 SUM,     0
   1432       0 K,       0
 //////  C O M M A N D  S U B R O U T I N E S  /////////////////
/
// SUBROUTINE SEND (ALIAS COM1)
/ REVISION -- NOVEMBER 25,1980
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- SEND DATA FROM CTL TO LISTENER(S)
/ PARAMETERS USED -- NONE PARAMETERS IN THE PARAMETER
/       TABLE ARE USED ONLY BY CTL-SEND
////////////////////////////////////////////////////////////////////
*1500
   1500       0 COM1,    0
   1501 3001714          JMS     @WCTL   /TRANSFER TABLE VALUES
   1502       0          TABLA
   1503      24          24
   1504 3001665          JMS     @MONITOR        /CATCH TABLE TRANSFER BYTE
   1505 3001665          JMS     @MONITOR        /WAIT UNTIL CTL IS DONE
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
   1506 1001500          JMP     @COM1
////////////////////////////////////////////////////////////////////////
// SUBROUTINE RECV (ALIAS COM2)
/ REVISION -- NOVEMBER 25,1980
```

```
/ AUTHOR   -- BARRETT,TB
/ PURPOSE -- TRANSFER DATA FROM TALKER TO CTL
/ PARAMETERS USED -- NONE
////////////////////////////////////////////////////////////////////////
   1507        0 COM2,   0
   1510 3001714          JMS     @WCTL
   1511        0         TABLA
   1512       24         24
   1513 3001665          JMS     @MONITOR
   1514 3001665          JMS     @MONITOR
   1515 1001507          JMP     @COM2
/
////////////////////////////////////////////////////////////////////////
/
//SUBROUTINE POLL (ALIAS COM4)
/ REVISION -- JANUARY 19,1981
/ AUTHOR   -- BARRETT,TB
/ PURPOSE -- CONDUCT A SERIAL POLL (THE STATUS BYTE IS TYPED)
/ PARAMETERS -- NONE
   1516        0 COM4,   0
   1517 3001714          JMS     @WCTL
   1520        0         TABLA
   1521       24         24
   1522 3001665          JMS     @MONITOR
   1523 3001665          JMS     @MONITOR
   1524 2405413          ACCM    TEMP     /STORE STATUS
   1525    5024          RASH    4        /GET READY FOR FIRST HEX DIGIT
   1526 3001666          JMS     @HEXT
   1527 2111413          MEMA    TEMP
   1530 3001666          JMS     @HEXT
   1531 2707540          MMOMZ   NCNT
   1532    1536          JMP     COM4E
   1533  110044          MEMA    (44
   1534 2405540          ACCM    NCNT
   1535 3001700          JMS     @CRLF
   1536 3001665 COM4E,   JMS     @MONITOR
   1537 1001516          JMP     @COM4
   1540        0 NCNT,   0
////////////////////////////////////////////////////////////////////////
/
// SUBROUTINE NICI  (ALIAS COM5)
/ REVISION -- DECEMBER 29,1980
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- WRITE DATA FROM NIC TO CTL
/ PARAMETERS USED --

/       (1) "I" FOR IMMEDIATE DATA (THE DATA TO SEND IS IN TABLE
/       LOCATIONS 2 => 10), OR THE STARTING ADDRESS IN NIC OF THE
        /BLOCK OF DATA TO BE SENT (CAN NOT BE "I" = 311 OCTAL).
```

```
/       (11) NO.OF DATA WORDS (1 BYTE/WORD) TO BE SENT OR
/       (12) NO. OF 256 BYTE BLOCKS TO BE TRANSFERRED IF (11)=0.
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
/       (18) -1 => DATA IS PACKED NIBBLES (WHEN UNPACKING ADD OCTAL 60
/                  TO TRANSFORM TO ASCII NUMBER.
/               0 => DATA IS PACKED
/               1 => DATA IS UNPACKED (5BYTES IN 2 WORDS)
/
/ NOTE -- THE STARTING ADDRESS IN (1) CAN BE -1 TO INDICATE THAT
/       THE UNPACKING PROCESS SHOULD CONTINUE FROM WHERE IT LEFT
/       OFF ON THE PREVIOUS CALL TO UNPF.
////////////////////////////////////////////////////////////////////////
   1541        0 COM5,   0
   1542  3001714          JMS     @WCTL
   1543        0          TABLA
   1544       24          24
   1545  3001665          JMS     @MONITOR
   1546  2110022          MEMA    DUM1
   1547  2405577          ACCM    FLAG5
   1550  2405663          ACCM    SCTLC
   1551  2110001          MEMA    NLIST   /IT IS IMMEDIATE MODE ?
   1552   462311          A-MZ    ("I
   1553     1555          JMP     #+2
   1554   110002          MEMA    (TABLA+2
   1555  2405600          ACCM    STADD5
   1556  2112013          MEMAZ   NDAT
   1557     1561          JMP     #+2
   1560     1563          JMP     #+3
   1561  2405576          ACCM    NBYTE5
   1562     1574          JMP     COM51
   1563  2112014          MEMAZ   NDATB
   1564  2162000          ZERZ
   1565     1601          JMP     COM53
   1566  2405664          ACCM    COUNTS  /DO ADD INSTEAD OF MULT.
   1567  2170000          ZERA
   1570   510400          A+MA    (400
   1571  2707664          MMOMZ   COUNTS
   1572     1570          JMP     #-2
   1573  2405576          ACCM    NBYTE5
   1574  3001701 COM51,   JMS     @UNPF
   1575     1650          SCTL
   1576        0 NBYTE5,  0
   1577        0 FLAG5,   0
   1600        0 STADD5,  0
   1601     4062 COM53,   CTLCF
   1602  3001665          JMS     @MONITOR
   1603  1001541          JMP     @COM5
////////////////////////////////////////////////////////////////////////
```

```
/
//SUB ROUTINE NICO (ALIAS COM6)
/ REVISION -- DECEMBER 29,1980
/ AUTHOR   -- BARRETT,TB
/ PURPOSE --READ DATA FROM CTL TO NIC
/ PARAMETERS USED --
/       (1) "I" FOR DATA TO BE STORED IN TABLE LOCATIONS 2 => 10,
/       OR STARTING ADDRESS FOR DATA STORAGE.
/       -1 MEANS USE LAST ADDRESS FROM PRIOR RUN
/       (11) NO. OF DATA WORDS TO BE TRANSFERRED OR
--------
/ CAMERA MAIN (NOVEMBER 5,1981)
/       (12) NO. OF 256 BYTE BLOCKS TO BE TRANSFERRED IF (11)=0.
/       (18) 1 => DO NOT PACK THE DATA
/            0 => PACK THE DATA (5 BYTES/2 WORDS)
   1604        0 COM6,   0
   1605  3001714         JMS     @WCTL
   1606        0         TABLA
   1607       24         24
   1610  3001665         JMS     @MONITOR
   1611  2110022         MEMA    DUM1
   1612  2405641         ACCM    FLAG6
   1613  2110001         MEMA    NLIST
   1614   462311         A-MZ    ("I
   1615     1617         JMP     #+2
   1616   110002         MEMA    (TABLA+2
   1617  2405642         ACCM    STADD6
   1620  2112013         MEMAZ   NDAT
   1621     1623         JMP     #+2
   1622     1625         JMP     #+3
   1623  2405640         ACCM    NBYT1
   1624     1636         JMP     COM61   /NOT 0
   1625  2112014         MEMAZ   NDATB
   1626  2162000         ZERZ
   1627     1644         JMP     COM63   /NOTHING TO TRANSFER
   1630  2405664         ACCM    COUNTS
   1631  2170000         ZERA
   1632   510400         A+MA    (400
   1633  2707664         MMOMZ   COUNTS
   1634     1632         JMP     #-2
   1635  2405640         ACCM    NBYT1
   1636  3001702 COM61,  JMS     @PAKF
   1637     2123         MONITA
   1640        0 NBYT1,  0
   1641        0 FLAG6,  0
   1642        0 STADD6, 0
   1643        0         0
   1644  3001665 COM63,  JMS     @MONITOR
   1645  1001604         JMP     @COM6
```

```
////////////////////////////////////////////////////////////////
/
/ SUBROUTINE SCTL
/ PURPOSE -- SEND BYTE TO CTL
/ THERE ARE 2 MODES OF OPERATION SET BY SCTLC. IF SCTLC=-1
/ THE BYTE IS SENT AS ASCII (60H IS ADDED TO ACC), OTHERWISE
/ IT IS SENT WITH NO CHANGE. SCTL MAY BE ABORTED
/ BY TYPING ANY CHARACTER ON THE TTY IN CASE THE CTL
/ GETS HUNG.
*1650
   1650        0 SCTL,   0
   1651 2123663          MPOZ     SCTLC
   1652 2162000          ZERZ
   1653  510060          A+MA     (60
   1654    4071          CTLWR
   1655    6064 SC1,     CTLSK
   1656    1660          JMP      SC2
   1657 1001650          JMP      @SCTL
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
   1660    6454 SC2,     TTYRF
   1661    1655          JMP      SC1
   1662 1001730          JMP      @ABORT
   1663        0 SCTLC,  0
 /
/ SCRATCH STORAGE
   1664        0 COUNTS, 0
/
/ ADDRESSES
        HEXTA=2330
        TABLA=0
        SENDFA=4242
        SCTLA=1650
        MONITA=2123
        WPNTD=3414
        PENDA=1770
/ EXTERNALS
   1665    2123 MONITOR,         MONITA
   1666    2330 HEXT,    HEXTA
   1667    2257 ECHO,    2257
   1670    2420 OPENW,   2420
   1671    2432 OPENR,   2432
   1672    2451 CLOSE,   2451
   1673    2470 WRITE,   2470
   1674    2511 READD,   2511
   1675    2605 PRTOCT,  2605
   1676    2650 UNP,     2650
   1677    2731 TYPE,    2731
   1700    2736 CRLF,    2736
```

```
   1701     2750 UNPF,    2750
   1702     3074 PAKF,    3074
   1703     3240 PKR,     3240
   1704     3310 DEC,     3310
   1705     3470 NIBBIN,  3470
   1706     3565 TEKHEX,  3565
   1707     4010 NICFIL,  4010
   1710     4070 SEARCH,  4070
   1711     4160 ZERTAB,  4160
   1712     4175 MULTP,   4175
   1713     4213 DIVDE,   4213
   1714     2136 WCTL,    2136
   1715     4313 GETFIL,  4313
   1716     4362 PAUSE,   4362
   1717     4371 OCT,     4371
/ DEFINITIONS AND COMMON
   1720     7770 OARG1,   7770
   1721     7771 OARG2,   7771
   1722     7772 OARG3,   7772
   1723     1770 PEND,    PENDA   /PAGE END FOR SCRATCH STORAGE
   1724   100000 BUFS,    100000
   1725     4311 SMODE,   SENDFA+47
   1726     4310 SWCNT,   SENDFA+46
   1727     4312 SWCNTO,  SENDFA+50
   1730     4700 ABORT,   4700
   1731     7760 SYSTRT,  7760
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
   1732     3223 PCOUNT,  3223
/ CONSTANTS
   1733   606251 PRI,     606251
   1734     3000 T3,      3000
   1735    40000 PA1,     40000   /PAUSE CONSTANTS
   1736    40000 PA2,     40000
   1737   100000 PA4,     100000
   1740     2000 BFSZ,    2000    /CTL BUFFER SIZE
 /////////// S E R V I C E   S U B R O U T I N E S   /////////////
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
ABORT     1730     APNT       551     BASE      140     BCNT      550
BFSZ      1740     BIN        615     BUFFER     16     BUFS     1724
CAMERA       2     CLOSE     1672     CODE      417     CODMS     650
COM1      1500     COM2      1507     COM4      1516    COM4E     1536
COM5      1541     COM51     1574     COM53     1601    COM6      1604
COM61     1636     COM63     1644     COM8       257    COMN         0
COMSTO     342     COUNT     1424     COUNTN     547    COUNTS    1664
CRLF      1700     CTABLE      24     CTL          1    CTLCF     4062
CTLRD    44064     CTLRDC   44066     CTLRS     4072    CTLSK     6064
CTLWR     4071     DATAH       16     DATAL       15    DEC       1704
```

DIVDE	1713	DUM1	22	DUM2	23	ECHO	1667
ECHO1	552	EOSC	12	EXTAN	6	FACT1	1453
FACTOR	730	FIL1	477	FIL2	416	FILE	452
FILM5	1415	FILM51	647	FILNM	1365	FILNM1	1347
FLAG	423	FLAG5	1577	FLAG6	1541	FLG4	1474
FLGM5	654	GETFIL	1715	HEXT	1661	HEXTA	1665
HORRES	5	INF	2	INT	676	INTER1	44
ISTART	1226	IT	1224	ITO	1422	J	1431
J1	1420	K	1432	L1	1433	LIS1P	3
LIS1S	3	LIS2P	4	LIS2S	5	LIS41	6
LIS3S	7	LISTEN	40	MARK	7	MEM	1447
MM1	466	MOD50	631	MOD53A	755	MOD53B	1000
MOD5E	1264	MOD5J0	1161	MOD5J1	1164	MOD5J2	1267
MOD5J3	1217	MOD5J4	1234	MOD6K	1223	MOD71	1337
MOD81	1357	MOD8A	1364	MOD8E	1405	MODE1	134
MODE2	216	MODE21	221	MODE3	263	MODE41	331
MODE4	365	MODE41	376	MODE4A	475	MODE4B	476
MODE44	504	MODE5	560	MODE51	701	MODE52	671
MODE53	675	MODE55	1073	MODE5A	1042	MODE5B	1054
MODE5B	1066	MODE5C	1045	MODE61	1110	MODE62	1113
MODE6	1265	MODE7	1325	MODE8	1345	MONITO	2123
MONITO	1665	MULTP	1712	N	1414	NBM5	652
NBM51	613	NBY1	436	NBY2	447	NBY4	473
NBY5	317	NBY6	1274	NBY8	1327	NBYE	1440
NBYTE	1425	NBYTE	421	NBYTE2	1576	NBYTE3	1577
NC	407	NCNT	1540	NDAT	13	NDATA	14
NDATB1	1416	NIB	515	NIB3	542	NIBBLE	1727
NICFIL	1707	NLIST	1	NNIB	513	NNIB2	1454
NW1	266	NW2	274	NW3	312	NW4	1
NWORD	1213	OARG1	1729	OARG2	1721	OARG3	1722
OCT	1717	OPENR	1671	OPENW	1657	OP	1
PA1	1735	PA2	1731	PA4	1727	PAGE	1
PAIS	1126	PAUSE	1716	PCOUNT	1736	PEN	1
PENDA	1770	PER	1763	PRI	1733	PRINT	1
PRINT1	1675	READ	1674	REM	1	REM	1
REM2	1202	RETAB	20	RETAB	17	RPT	1
SC1	1655	SC2	1660	SCTL	1652	SEEK	1
SETLC	1663	SEARCH	1715	SEND1A	1734	SEND2	1
SLICE	13	SLICE5	15	SLICE1	14	SM	1
STADD5	1600	STADD6	1642	START	21	SW	1
SWCNT	1726	SWCNT0	1727	SYSTEM	1741	T	1
TABLA	0	TALK	100	TALK1	13	TAPE	1
TEKHEX	1706	TEMP	1414	TPEOF	1	TPER	1
UNP	1676	UNPF	1721	VAL	4	VAL	1
VALM5	601	VALM51	612	VALM52	621	VALM6	1

/ CAMERA MAIN (NOVEMBER 6, 1981)

VIDIN	10	VIDIN1	12	VIDIN2	11	WAIT	1
WCTL	1714	WORD	545	WRNT	1	WAIT	1

```
WRITE      1673      XCOORD        3      ZERTAB     1711
$
--------
/ CAMERA MAIN (NOVEMBER 6,1981)
W
*
```

```
#
--------
 / FUNCTION ERROR
// FUNCTION ERROR
/ REVISION -- NOVEMBER 6,1981
/ PURPOSE -- ERROR IS THE "ABORT" ROUTINE WHICH IS ENTERED WHEN THE SYSTE
/ "HUNG". THE USER CAN TYPE ANY CHARACTER ON THE CONSOLE WHICH CAUSES
/ ERROR TO START. ERROR FIRST CAUSES NICCTL TO INTERRUPT AND TO EXECUTE
/AN INTERRUPT SUBROUTINE WHICH STORES THE "STATE" OF NICFIL IN RAM STARTI
/ AT 80C AND TO THEN REINITIALIZE THE BUS AND WAIT FOR COMMANDS FROM NIC.
/ERROR GATHERS THE STATE DATA PLUS THE CONTENTS OF THE CTL COMMAND TABLE
/AND PRINTS THIS DATA ALONG WITH IDENTIFIERS AS FOLLOWS:
/
/       ROW 1                                   ROW 2
/       -----                                   -----
/
/       0 = COMM                                L = L REGISTER
/       1 = NLIST                               H = H REGISTER
/       2 = LIS1P                               PC= PROGRAM COUNTER
/       3 = LIS1S                               SP= STACK POINTER
/       4 = LIS2P                               A = A REGISTER
/       5 = LIS2S                               C = C REGISTER
/       6 = LIS3P                               B = B REGISTER
/       7 = LIS3S                               E = E REGISTER
/       8 = TALKP                               D = D REGISTER
/       9 = TALKS                               0 = 8291 INT1
/       A = EOSC                                1 = 8291 INT2
/       B = NDAT                                2 = 8291 ADRST
/       C = NDATB                               3 = 8291 ADR01
/       D = DATADD(LO)                          4 = 8291 EOSR
/       E = DATADD(HI)                          5 = 8292 INTST
/       F = DUM1H                               6 = 8292 ERROR FLAG
/       0 = DUM1H                               7 = 8292 CONTROLLER STATUS
/       1 = MESS                                8 = 8292 BUS STATUS
/       2 = STAT1
/       3 = STAT2
/       4 = RETADD(LO)
/       5 = RETADD(HI)
/       6 = LCOMM
/       7 = MAXBLK
/       8 = STACKP(LO)
/       9 = STACKP(HI)
/       A = COUNT(LO)
/       B = COUNT(HI)
////////////////////////////////////////////////////////////////////////////////
/
        CTLSK=6064
        CTLRDC=44066
```

```
        CTLCF=4062
        CTLRS=4072
        CTLWR=4071
*2020
   2020    4071 ERROR,  CTLWR   /CAUSE INTERRUPT
/ PRINT THE FIRST HEADER TO GIVE CTL TIME TO INITIALIZE.
   2021 3000376          JMS     @CRLF
   2022 3000375          JMS     @UNP
   2023      70          70
--------
// FUNCTION ERROR
   2024       1          1
   2025  200021          TEXT %0 1
   2026    2200    2
   2027  230024   3 4
   2030    2500    5
   2031  260027   6 7
   2032    3000    8
   2033  310041   9 A
   2034    4200    B
   2035  430044   C D
   2036    4500    E
   2037  460020   F 0
   2040    2100    1
   2041  220023   2 3
   2042    2400    4
   2043  250026   5 6
   2044    2700    7
   2045  300031   8 9
   2046    4100    A
   2047  420077   B %
   2050 3000403          JMS     @ZERTAB         /ZERO THE COMMAND TABLE
   2051       0          TABLA
   2052  110057          MEMA    (57
   2053 3404413          ACCM    @NDAT
   2054  110010          MEMA    (10
   2055 3404414          ACCM    @DATAH
   2056 2110407          MEMA    BUFS
   2057 3404415          ACCM    @NLIST
   2060  110006          MEMA    (6
   2061 3404416          ACCM    @COMN
   2062 3000404          JMS     @COM6
   2063  110034          MEMA    (34
   2064 2404173          ACCM    PRHEX1
   2065 2110407          MEMA    BUFS
   2066 2404175          ACCM    PRHEX2
   2067 2000164          JMS     PRHEX
   2070 3000376          JMS     @CRLF
   2071 3000375          JMS     @UNP
```

```
   2072       51          51
   2073        1          1
   2074   540050          TEXT %L H
   2075     6043    PC
   2076       63     S
   2077   600000   P
   2100   410043   A C
   2101     4200    B
   2102   450044   E D
   2103     2000    0
   2104   210022   1 2
   2105     2300    3
   2106   240025   4 5
   2107     2600    6
   2110   770000   %
   2111   110022          MEMA     (22      /UNPACK THE NEXT 2? VALUES
   2112 2404173           ACCM     PRHEX1
--------
//  FUNCTION ERROR
   2113 2144175           MONM     PRHEX2
   2114 2000164           JMS      PRHEX
   2115 3000376           JMS      @CRLF
   2116 1000402           JMP      @ABORT
/
//FUNCTION MONITOR
/ REVISION -- NOVEMBER 2,1981
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- "MONITOR" INPUT FROM CTL. IT READS DATA FROM CTL
/       AND JUMPS TO ERROR IF SRVC BIT IS SET. OTHERWISE IT
/       RETURNS THE BYTE READ IN ACC.
/       IF THE USER TYPES A CHARACTER, MONITOR WILL JUMP TO ERROR.
/
        SRVC=400
*2123
   2123        0 MONITOR,          0
   2124     6064 MON1,    CTLSK
   2125      132           JMP      MON2
   2126    44066           CTLRDC
   2127     2400           ANDZ     (SRVC
   2130 2000020           JMS      ERROR
   2131 1000123           JMP      @MONITOR
   2132     6454 MON2,    TTYRF
   2133      124           JMP      MON1
   2134    44453           RDTTY
   2135       20           JMP      ERROR
/
//SUBROUTINE WCTL(STADD,NBYTES)
/ REVISION -- JANUARY 22,1981
/ AUTHOR   -- BARRETT,TB
```

```
/ PURPOSE   -- TRANSFER NBYTES OF DATA FROM NIC MEMORY STARTING AT
/       ADDRESS STADD TO CTL
/ PARAMETERS --
/       STADD   STARTING ADDRESS OF DATA BLOCK IN NIC
/       NBYTES  SIZE OF DATA BLOCK IN BYTES (1 BYTE/NIC WORD
/
   2136       0 WCTL,   0
   2137 3164417         ZERM    @SCTLC
   2140 3110136         MEMA    @WCTL
   2141 2404155         ACCM    POINT    /POINTS TO DATA BLOCK
   2142 2124136         MPOM    WCTL     /GET COUNT
   2143 3110136         MEMA    @WCTL
   2144 2404156         ACCM    COUNT
   2145 2124136         MPOM    WCTL     /SET FOR RETURN FROM WCTL
   2146 3110155 WCTL1,  MEMA    @POINT   /GET NEXT DATUM
   2147 3000406         JMS     @SCTL
   2150 2124155         MPOM    POINT
   2151 2706156         MMOMZ   COUNT
   2152     146         JMP     WCTL1
   2153    4062         CTLCF   /CLEAR DONE ON LAST WRITE
   2154 1000136         JMP     @WCTL
   2155       0 POINT,  0
   2156       0 COUNT,  0
/
/
--------
// FUNCTION ERROR
   2157       0 RCTL,   0
   2160    6064         CTLSK
   2161     160         JMP     #-1
   2162   44066         CTLRDC
   2163 1000157         JMP     @RCTL


/ SUBROUTINE PRHEX
/ PURPOSE -- THIS IA SUBROUTINE TO SET UP CALLS TO UNPF. IT IS
/       USED WITH ERROR.
   2164       0 PRHEX,  0
   2165 3024410         ONEM    @SMODE
   2166  110050         MEMA    (50      /SO NO CR LF
   2167 3404411         ACCM    @SWCNT
   2170 3404412         ACCM    @SWCNTO
   2171 3000405         JMS     @UNPF
   2172    4242         SENDFA
   2173       0 PRHEX1, 0       /NO. OF BYTES TO UNPACK
   2174       0         0       /UNPACK
   2175       0 PRHEX2, 0       /SET TO 100000 FIRST CALL, -1 NEXT
   2176 1000164         JMP     @PRHEX
/
```

```
/

*2257
   2257        0 ECHO,   0
   2260  2000271         JMS     READ
   2261   462221         A-MZ    (221     /^Q
   2262  2162000         ZERZ
   2263  1000402         JMP     @ABORT
   2264   462207         A-MZ    (207     /^G
   2265  2162000         ZERZ
   2266  1000420         JMP     @CALLS
   2267  3000401         JMS     @TYPE
   2270  1000257         JMP     @ECHO
   2271        0 READ,   0
   2272     6454         TTYRF
   2273      272         JMP     #-1
   2274    44453         RDTTY
   2275  1000271         JMP     @READ



// FUNCTION HEXT
/ PURPOSE -- CONVERT 4 BIT HEX (LEFT JUSTIFIED) IN ACC TO
/            ASCII AND TYPE IT.
HEXT,   0
        ANDA    (17
        A+MA    (260
        ACCM    TEMP
        M-AA    (271
        SKIP    AC19
        JMP     #+3
        MEMA    (7
        A+MM    TEMP
        MEMA    TEMP
        JMS     @TYPE
        JMP     @HEXT
/
/ TEMP STORAGE AND TABLE
TEMP,0
```

```
        SENDFA=4242
        TABLA=0
/
/
/ EXTERNALS
   2375~~44~~  2650 UNP,     2650
   2376     2736 CRLF,    2736
   2377     2605 PRTOCT,  2605
   2400     7600 SYSTRT,  7600
   2401     2731 TYPE,    2731
   2402     4700 ABORT,   4700
   2403     4160 ZERTAB,  4160
   2404     1604 COM6,    1604
   2405     2750 UNPF,    2750
   2406     1650 SCTL,    1650
/ DEFINITIONS AND COMMON
   2407   100000 BUFS,    100000
   2410     4311 SMODE,   SENDFA+47
   2411     4310 SWCNT,   SENDFA+46
   2412     4312 SWCNTO,  SENDFA+50
   2413       13 NDAT,    TABLA+13
   2414       16 DATAH,   TABLA+16
   2415        1 NLIST,   TABLA+1
   2416        0 COMN,    TABLA+0
--------
// FUNCTION ERROR
   2417     1663 SCTLC,   1663
   2420       71 CALLS,   71
--------
// FUNCTION ERROR
ABORT     2402     BUFS      2407     CALLS     2420     COM6      2404
COMN      2416     COUNT     2156     CRLF      2376     CTLCF     4062
```

```
CTLRDC   44066     CTLRS    4072     CTLSK    6064     CTLWR    4071
DATAH     2414     ECHO     2257     ERROR    2020     HC       2351
HC1       2352     HEXT     2330     HP       2353     HT       2354
HTOP      2355     HTYPE    2345     K17      2350     MON1     2124
MON2      2132     MONITO   2123     NDAT     2413     NLIST    2415
POINT     2155     PRHEX    2164     PRHEX1   2173     PRHEX2   2175
PRTOCT    2377     RCTL     2157     READ     2271     SCTL     2406
SCTLC     2417     SENDFA   4242     SMODE    2410     SRVC      400
SWCNT     2411     SWCNTO   2412     SYSTRT   2400     TABLA       0
TYPE      2401     UNP      2375     UNPF     2405     WCTL     2136
WCTL1     2146     ZERTAB   2403
$
--------
// FUNCTION ERROR
*
```

```
#
--------
E// I/O SUBROUTINES OPENW,OPENR,CLOSE,WRITE,READD
 // I/O SUBROUTINES OPENW,OPENR,CLOSE,WRITE,READD
 //SUBROUTINE OPENW
/ REVISION -- JANUARY 26,1981
 /  AUTHOR    - BARRETT,TB
 /  PURPOSE  -- OPENS A FILE (FOR WRITING) BY LOCATING THE NEXT AVAILABLE
 /              TRACK AND AMOUNT OF SPACE AAILABLE.
 /
 /  PARAMETERS -- NONE. TRACK AND SPACE ARE RETURNED IN OARG1
 /                (7770) AND OARG2 (7771) RESPECTIVELY,IF OARG2=0.
/       IF OARG2 IS SET TO THE NO. OF WORDS IN THE FILE,OPENW
/       WILL FIND THE FIRST AVAILABLE SPACE.
/       NOTE - SET OARG1 TO 0 BEFORE CALLING.
*2420
   2420       0 OPENW,  0
   2421 2000527         JMS     DIRFIN
   2422 3144560         MONM    @DISOLV
   2423 3000555         JMS     @DIRFUN
   2424       1         1
   2425       2         2
   2426    2567         NOFIL   /POINTS TO A VALUE OF 0
   2427 2410000         ACCA
   2430 2000545         JMS     DIROUT
   2431 1000420         JMP     @OPENW
/
// SUBROUTINE OPENR(FILNAM)
/ REVISION -- NOVEMBER 12,1981
/ AUTHOR   --BARRETT,TB
/ PURPOSE  -- OPENS A FILE FOR READING BY RETURNING THE STARTING
/       TRACK,FILE SIZE AND LOCATION IN CORE FOR
/       STORAGE (AS GIVEN BY THE DIRECTORY)
/ ARGUMENTS --
/       FILNAM - 2 WORD PACKED FILE NAME OF THE FILE TO BE OPENNED.
/ LOCATIONS OARG1,OARG2,OARG3 CONTAIN TRACK,SIZE AND CORE LOCATION FOR
/ THE FILE RESPECTIVELY. IF THE FILE IS NOT FOUND, OARG1 CONTAINS -1.
   2432       0 OPENR,  0
   2433 3110432         MEMA    @OPENR  /FILENAME
   2434 2404565         ACCM    FILNM
   2435 2124432         MPOM    OPENR
   2436 2410000         ACCA
   2437 2410000         ACCA
   2440 2124432         MPOM    OPENR   /SET RETURN ADDRESS
   2441 2000527         JMS     DIRFIN
   2442 3000555         JMS     @DIRFUN
   2443       1         1
   2444       2         2
```

```
   2445    2565          FILNM
   2446 3144562          MONM    @OARG1  /FILE DOES NOT EXIST
   2447 2000545          JMS     DIROUT  /RESTORE
   2450 1000432          JMP     @OPENR
 /
 //  SUBROUTINE CLOSE(FILNAM)
 /  PURPOSE -- ADD A FILE TO THE DIRECTORY
/       BY SETTING OARG2 (7771) TO 0 THE FILE WILL BE DELETED
 /  PARAMETERS -
--------
E// I/O SUBROUTINES OPENW,OPENR,CLOSE,WRITE,READD
 /  FILNAM - 6 CHAR. (PACKED FORM) FILE NAME (2 WORDS)
 /     CONTROL RETURNS AFTER THE FILE NAME
 /  BEFORE CALLING PUT THE STARTING TRACK IN 7770 AND THE
 /  FILE SIZE (WORDS) IN 7771. THE CODE ADDRESS CAN BE PUT INTO
/       OARG3 AND THE STARTING ADDRESS IN SYSTRT.
   2451       0 CLOSE,  0
   2452 3110451          MEMA    @CLOSE  /TRANSFER FILENAME
   2453 2404565          ACCM    FILNM
   2454 2124451          MPOM    CLOSE
   2455 2410000          ACCA
   2456 2410000          ACCA
   2457 2124451          MPOM    CLOSE
   2460 2000527          JMS     DIRFIN
   2461 3000555          JMS     @DIRFUN
   2462       1          1
   2463       1          1
   2464    2565          FILNM
   2465 2410000          ACCA
   2466 2000545          JMS     DIROUT
   2467 1000451          JMP     @CLOSE
 /
 //  SUBROUTINE WRITE(IT,SIZE,ISTART)
 / PURPOSE -- SIMPLE WRITE TO DISK USING DEMON II DISK
 /  PARAMETERS --
 /  IT - STARTING TRACK
 /  SIZE - NO. PF WORDS IN BUFFER (STARTS AT ISTART)
/   ISTART - STARTING ADDRESS OF BLOCK TO TRANSFER.
   2470       0 WRITE,  0
   2471 3110470          MEMA    @WRITE
   2472 2510510          A+MA    DNO
   2473 2404504          ACCM    IT
   2474 2124470          MPOM    WRITE
   2475 3110470          MEMA    @WRITE
   2476 2404505          ACCM    SIZE
   2477 2124470          MPOM    WRITE
   2500 3110470          MEMA    @WRITE
   2501 2404506          ACCM    ISTART
   2502 2124470          MPOM    WRITE   /RETURN ADDRESS
```

```
   2503 3000556          JMS     @DISK
   2504        0 IT,     0
   2505        0 SIZE,   0
   2506        0 ISTART, 0
   2507 1000470          JMP     @WRITE
   2510  100000          DNO,    100000
// SUBROUTINE READD(IT,SIZE)
/ REVISION -- NOVEMBER 29,1980
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- READ TRACK IT OF SIZE WORDS INTO BUFFER
/               STARTING AT 100000. NOTE THAT IF SIZE IS GREATER
/               THAN A TRACK, MORE THAN 1 TRACK WILL BE READ.
   2511        0 READD,  0
   2512 3110511          MEMA    @READD
   2513 2510510          A+MA    DNO     /ADD THE DISK NO.
   2514 2404523          ACCM    ITT
   2515 2124511          MPOM    READD
--------
E// I/O SUBROUTINES OPENNW,OPENR,CLOSE,WRITE,READD
   2516 3110511          MEMA    @READD
   2517 2404524          ACCM    SIZZ
   2520 2124511          MPOM    READD   /SET RETURN
   2521 2170000          ZERA    /SIGNALS READ
   2522 3000556          JMS     @DISK
   2523        0 ITT,    0
   2524        0 SIZZ,   0
   2525  100000          100000
   2526 1000511          JMP     @READD
 /
   2527        0 DIRFIN, 0       /READ OUT 3000-7600,READ IN DIRFUN
   2530 2030000          ONEA
   2531 3000556          JMS     @DISK
   2532  100001          100001
   2533    4600          4600
   2534    3000          3000
   2535 3174557          ZERMA   @DERRF
   2536 3000556          JMS     @DISK
   2537  100007          100007
   2540     600          600
   2541    7000          7000
   2542 2410000          ACCA
   2543 3164561          ZERM    @DEVDET
   2544 1000527          JMP     @DIRFIN
 //
   2545        0 DIROUT, 0       /READ BACK 3000-7600
   2546 3174557          ZERMA   @DERRF
   2547 3000556          JMS     @DISK
   2550  100001          100001
   2551    4600          4600
```

```
   2552     3000           3000
   2553 2410000            ACCA
   2554 1000545            JMP      @DIROUT
 /
 / DEMON II REFERENCES
   2555     7000 DIRFUN,   7000
   2556     7612 DISK,     7612
   2557     7704 DERRF,    7704
   2560     7751 DISOLV,   7751
   2561     7764 DEVDET,   7764
   2562     7770 OARG1,    7770
   2563     7771 OARG2,    7771
   2564     7772 OARG3,    7772
/ SCRATCH STORAGE
   2565        0 FILNM,    BLOCK   2
   2567        0 NOFIL,    0
--------
E// I/O SUBROUTINES OPENW,OPENR,CLOSE,WRITE,READD
CLOSE      2451      DERRF      2557      DEVDET     2561      LIRFIN     2527
DIRFUN     2555      DIROUT     2545      DISK       2556      DISOLV     2560
DNO        2510      FILNM      2565      ISTART     2506      IT         2504
IIT        2523      NOFIL      2567      OARG1      2562      OARG2      2563
OARG3      2564      OPENR      2432      OPENW      2420      READ       2511
SIZE       2505      SIZZ       2524      WRITE      2470
$
--------
E// I/O SUBROUTINES OPENW,OPENR,CLOSE,WRITE,READD
*
```

```
#
--------
  //   FUNCTION PRTOCT(X)
  //   FUNCTION PRTOCT(X)
/ REVISION -- JANUARY 22,1981
 /   AUTHOR    -- BARRETT,TB
 /   PURPOSE   -- PRINT THE OCTAL VALUE OF THE CONTENTS OF ACC
 /
 /
*2605
   2605        0 PRTOCT,  0
   2606     5042          LLSH    2
   2607 2405067           ACCM    TEMP
   2610    10003          ANDA    (3
   2611   510260          A+MA    (260
   2612 2000731           JMS     TYPE
   2613   110007          MEMA    (7        /SET COUNTER
   2614 2405064           ACCM    COUNT
   2615 2707064 PRT01,    MMOMZ   COUNT
   2616      620          JMP     #+2
   2617      627          JMP     PRT02
   2620 2111067           MEMA    TEMP
   2621     5043          LLSH    3
   2622 2405067           ACCM    TEMP
   2623    10007          ANDA    (7
   2624   510260          A+MA    (260
   2625 2000731           JMS     TYPE
   2626      615          JMP     PRT01
   2627   110215 PRT02,   MEMA    (215
   2630 2000731           JMS     TYPE
   2631   110212          MEMA    (212
   2632 2000731           JMS     TYPE
   2633 1000605           JMP     @PRTOCT
 // SUBROUTINE UNP(NC,INDIC,TEXT)
/REVISION --NOVEMBER 22,1980
/AUTHOR    -- BARRETT,TB
/PURPOSE    -- UNPAK PACKED ASCII AND SENDS TO TTY FOR PRINTING.
/        AN OPTIONAL CR/LF IS SENT ALSO.
/
/PARAMETERS --
/        NC       NO. OF CHARACTERS IN THE PACKED TEXT. IF 0, THE
/        TEXT IS ASSUMED TO BE TERMINATED WITH ?? (%) AND NC IS
/        RETURNED AS THE NO. OF TEXT CHARACTERS (NOT INCLUDING %)
/        INDIC    0 => NO CR/LF, 1 => CR/LF AT END OF TEXT.
/        TEXT     THE PACKED TEXT.
/
*2650
   2650     2??5 UNP      ,0
```

```
 2651 2110650          MEMA    UNP     /STORE ADDRESS OF NC
 2652 2404727          ACCM    NC
 2653 2124650          MPOM    UNP     /STORE INDIC
 2654 3110650          MEMA    @UNP
 2655 2405303          ACCM    INDIC
 2656 2024730          ONEM    INDIX   /SET PRINT/NOPRINT INDICATOR
 2657 2164726          ZERM    NCC     /SET CHARACTER COUNTER TO 0
 2660 2025064          ONEM    COUNT   /INITIALIZE 1,2,3 COUNTER
 2661 2707064 LOOP,    MMOMZ   COUNT   /DECREMENT COUNTER. IF 0 GET NEXT W
--------
//  FUNCTION PRTOCT(X)
 2662      671          JMP     L1      /IF NOT 0, TYPE CHARACTER
 2663   110003          MEMA    (3      /REINITIALIZE COUNTER
 2664 2405064          ACCM    COUNT
 2665 2124650          MPOM    UNP     /POINT TO NEXT WORD IN TEXT
 2666 3110650          MEMA    @UNP    /GET WORD AND SHIFT IT
 2667     5050          LLSH    8
 2670 2405230          ACCM    WORD    /STORE IT FOR EXAMPLE W
 2671    10077 L1,     ANDA    (77     /MASK 6 LSDS
 2672   462077         A-MZ    (77     /CHECK FOR END OF TEXT
 2673      675          JMP     #+2
 2674      720          JMP     END
 2675 2102727          MEMZ    @NC
 2676      700          JMP     #+2
 2677      706          JMP     L3
 2700 2405067          ACCM    TEMP
 2701 2110727          MEMA    @NC
 2702 2467706          A-MZ    NCC     /CHECK TO SEE IF NC=6, IF SO, SENT
 2703      705          JMP     #+2
 2704 1624731          ZERM    INDIX   /IF INDIX=0, CHARACTER ARE N
 2705 2111067          MEMA    TEMP
 2706 2124730 L3,      MEMZ    INDIC   /IF 0, DON'T PRINT
 2707      711          JMP     #+2
 2710      714          JMP     #+4
 2711 2467706          MPOM    NCC
 2712 2410240          A+MA    (240    /CONVERT TO CHARACTER
 2713 4002741          JMS     TYPE
 2714 2111067          MEMA    WORD
 2715     5050          LLSH    8       /SHIFT NEXT BYTE
 2716 2405230          ACCM    WORD
 2717      661          JMP     LOOP    /GET NEXT BYTE
 2720 1624731 END,     MEMAZ   INDIX
 2721 4002741          JMS     TYPE
 2722 1624731          MEMZ    TYP     /IF 0 RETURN
 2723 2111067          MEMA    NCC
 2724 2467727          ACCM    @NC     PRINT NC
 2725 4000000          JMP     @RTN
 2726              NC,
 2727              RTN,
```

```
   2730        0 INDIX,  0
/
/////// SUBROUTINE TYPE //////
   2731        0 TYPE,   0
   2732     6444         TTYPF
   2733      732         JMP      #-1
   2734     4443         PRTTY
   2735 1000731          JMP      @TYPE
/
////// SUBROUTINE CRLF  //////////
   2736        0 CRLF,   0
   2737   110212         MEMA     (212
   2740 2000731          JMS      TYPE
   2741   110215         MEMA     (215
   2742 2000731          JMS      TYPE
   2743 2000731          JMS      TYPE
   2744 1000736          JMP      @CRLF
--------
  //  FUNCTION PRTOCT(X)
 // SUBROUTINE UNPF(SENDF,NBYTES,FLAG,STARTA)
/ REVISION -- DECMEBER 24,1980
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- TRANSFER DATA FROM CORE TO A DESTINATION SPECIFIED BY
/       SENDF. THE DATA MAY BE UNPACKED IN THE PROCESS.
/
/ PARAMETERS --
/       SENDF - ENTRY POINT FOR ACCEPTING A WORD IN ACC (E.G. TYPE)
/       NBYTES - NO. OF BYTES TO BE TRANSFERRED. (FOR PACKED DATA
/                THERE ARE 2.5 BYTES/NIC WORD)
/       FLAG -   1 => DO NOT UNPACK
/                0 => UNPACK
/                -1 => UNNIBBLE (5 NIBBLES/WORD)
/       STARTA - STARTING ADDRESS AT WHICH TO OBTAIN DATA. IF
/       SET TO -1, UNPF WILL USE THE POINTER FROM THE PREVIOUS CALL
*2750
   2750        0 UNPF,   0
   2751 3110750          MEMA     @UNPF
   2752 2405063          ACCM     SENDF
   2753 2124750          MPOM     UNPF
   2754 3110750          MEMA     @UNPF
   2755 2405064          ACCM     COUNT
   2756 2124750          MPOM     UNPF
   2757 3110753          MEMA     @UNPF
   2760 2405065          ACCM     UNPFLG
   2761   110017         MEMA     (17
   2762 2124065          MPOZ     UNPFLG
   2763   110377         MEMA     (377
   2764 2405067          ACCM     MASK
   2765 2124750          MPOM     UNPF
```

```
2766 3110750          MEMA    @UNPF
2767    5144          EXCT    AC19    /IF NEG. THEN DONT INITIALIZE
2770     773          JMP     UNPFX
2771 2405066          ACCM    POINT
2772     775          JMP     UNPF1
2773 2703065 UNPFX,   MMOZ    UNPFLG  /TEST FLAG
2774    1014          JMP     UNPFZ
2775 3111066 UNPF1,   MEMA    @POINT
2776 2125066          MPOM    POINT
2777 2703065          MMOZ    UNPFLG
3000    1005          JMP     UNPFY
3001 3001063          JMS     @SENDF
3002 2707064          MMOMZ   COUNT
3003     775          JMP     UNPF1
3004    1061          JMP     UNPFE
3005 2405067 UNPFY,   ACCM    TEMP
3006  110006          MEMA    (6
3007 2405070          ACCM    BCOUNT
3010    1014          JMP     #+4
3011 2707064 UNPF2,   MMOMZ   COUNT
3012    1014          JMP     #+2
3013    1061          JMP     UNPFE
3014 2717070 UNPFZ,   MMOMAZ  BCOUNT
3015    1017          JMP     #+2
3016     775          JMP     UNPF1
--------
//  FUNCTION PRTOCT(X)
3017  462005          A-MZ    (5
3020    1031          JMP     UNPF3
3021 2111067 UNPF22,  MEMA    TEMP
3022 2123065          MPOZ    UNPFLG
3023    5044          LLSH    4
3024    5044          LLSH    4
3025 2405067          ACCM    TEMP
3026 2011073 UNPF21,  ANDA    MASK
3027 3001063          JMS     @SENDF
3030    1011          JMP     UNPF2
3031  462004 UNPF3,   A-MZ    (4
3032    1043          JMP     UNPF4
3033 2123065 UNPF33,  MPOZ    UNPFLG
3034 2162000          ZERZ
3035    1021          JMP     UNPF22
3036 2111067          MEMA    TEMP
3037  405024          RISH    4
3040   10360          ANDA    (360
3041 2405071          ACCM    TEMP1
3042    1021          JMP     UNPF22
3043  462003 UNPF4,   A-MZ    (3
3044    1060          JMP     UNPF5
```

```
   3045 2123065          MPOZ    UNPFLG
   3046 2162000          ZERZ
   3047    1021          JMP     UNPF22
   3050 3111066          MEMA    @POINT
   3051 2125066          MPOM    POINT
   3052    5044          LLSH    4
   3053 2405067          ACCM    TEMP
   3054   10017          ANDA    (17
   3055 2511071          A+MA    TEMP1
   3056 3001063          JMS     @SENDF
   3057    1011          JMP     UNPF2
   3060    1021 UNPF5,   JMP     UNPF22
   3061 2124750 UNPFE,   MPOM    UNPF     /RETURN
   3062 1000750          JMP     @UNPF
   3063       0 SENDF,   0
   3064       0 COUNT,   0
   3065       0 UNPFLG,  0
   3066       0 POINT,   0
   3067       0 TEMP,    0
   3070       0 BCOUNT,  0
   3071       0 TEMP1,   0
   3072       0 TEMP2,   0
   3073       0 MASK,    0
////////////////////////////////////////////////////////////////////////
/
// SUBROUTINE PAKF(RECVF,NBYTES,FLAG,STARTA,NBYTR)
/ REVISION -- DECEMBER 31,1980
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- TRANSFER DATA GIVEN BY RECVF TO CORE.
/       THE DATA MAY BE PACKED IN THE PROCESS. (THIS IS THE INVERSE
/       OF UNPF). USE OF ARGUMENTS IS THE SAME AS IN UNPF EXCEPT-
/       RECVF GETS A DATA BYTE AND GIVES IT TO PACKF VIA ACC. NOTE
/       THAT IT MAY BE NECESSARY FOR RECVF TO CONTROL THE NUMBER OF
--------
  //  FUNCTION PRTOCT(X)
/       BYTES TRANSFERRED BY STICKING A 1 IN COUNT1 WHEN THE LAST
/       BYTE HAS BEEN RECEIVED (E.G. AN EOF MARK IS DETECTED)
/       NBYTR - NO. OF BYTES RECEIVED.
////////////////////////////////////////////////////////////////////////
   3074       0 PAKF,    0
   3075 3111074          MEMA    @PAKF
   3076 2405222          ACCM    RECVF
   3077 2125074          MPOM    PAKF
   3100 3111074          MEMA    @PAKF
   3101 2405223          ACCM    COUNT1
   3102 2125223          MPOM    COUNT1
   3103 2125074          MPOM    PAKF
   3104 3111074          MEMA    @PAKF
   3105 2405225          ACCM    PAKFLG
```

```
3106 2125074           MPOM    PAKF
3107 3111074           MEMA    @PAKF
3110    5144           EXCT    AC19
3111    1117           JMP     PAKFX
3112 2405224           ACCM    POINT1
3113  110006           MEMA    (6
3114 2405231           ACCM    BCNT
3115 2165226           ZERM    NBYTES
3116 2165230           ZERM    WORD
3117 2703225 PAKFX,    MMOZ    PAKFLG
3120    1131           JMP     PAKF2
3121 2707223 PAKF1,    MMOMZ   COUNT1
3122    1124           JMP     #+2
3123    1215           JMP     PAKFF
3124 3001222           JMS     @RECVF
3125 2125226           MPOM    NBYTES
3126 3405224           ACCM    @POINT1
3127 2125224           MPOM    POINT1
3130    1121           JMP     PAKF1
3131 2707223 PAKF2,    MMOMZ   COUNT1
3132    1134           JMP     #+2
3133    1210           JMP     PAKFE
3134 2707231           MMOMZ   BCNT
3135    1140           JMP     #+3
3136  110005           MEMA    (5
3137 2405231           ACCM    BCNT
3140 3001222           JMS     @RECVF
3141 2405227           ACCM    TMP
3142 2125226           MPOM    NBYTES
3143 2111231           MEMA    BCNT
3144  462005           A-MZ    (5
3145    1152           JMP     PAKF3
3146 2111227 PAKF22,   MEMA    TMP
3147    5070           RLSH    10
3150 2505230           A+MM    WORD
3151    1131           JMP     PAKF2
3152  462004 PAKF3,    A-MZ    (4
3153    1160           JMP     PAKF4
3154 2111227           MEMA    TMP
3155    5044           LLSH    4
3156 2505230           A+MM    WORD
--------
  //  FUNCTION PRTOCT(X)
3157    1131           JMP     PAKF2
3160  462003 PAKF4,    A-MZ    (3
3161    1174           JMP     PAKF5
3162 2111227           MEMA    TMP
3163  405024           RISH    4
3164 2511230           A+MA    WORD
```

```
   3165 3405224          ACCM    @POINT1
   3166 2125224          MPOM    POINT1
   3167 2111227          MEMA    TMP
   3170   10017          ANDA    (17
   3171    5064          RLSH    4
   3172 2405230          ACCM    WORD
   3173    1131          JMP     PAKF2
   3174  462002 PAKF5,   A-MZ    (2
   3175    1202          JMP     PAKF6
   3176 2111227          MEMA    TMP
   3177    5050          LLSH    10
   3200 2505230          A+MM    WORD
   3201    1131          JMP     PAKF2
   3202 2111227 PAKF6,   MEMA    TMP
   3203 2511230          A+MA    WORD
   3204 3405224          ACCM    @POINT1
   3205 2125224          MPOM    POINT1
   3206 2165230          ZERM    WORD
   3207    1131          JMP     PAKF2
   3210 2703231 PAKFE,   MMOZ    BCNT
   3211 2162000          ZERZ
   3212    1215          JMP     #+3
   3213 2111230          MEMA    WORD
   3214 3405224          ACCM    @POINT1
   3215 2125074 PAKFF,   MPOM    PAKF
   3216 2111226          MEMA    NBYTES
   3217 3405074          ACCM    @PAKF
   3220 2125074          MPOM    PAKF
   3221 1001074          JMP     @PAKF   /NOTE THAT WE HAVE TO STORE THE LAST
/                        WHICH MAY BE LATER OVER WRITTEN.
   3222        0 RECVF,  0
   3223        0 COUNT1, 0
   3224        0 POINT1, 0
   3225        0 PAKFLG, 0
   3226        0 NBYTES, 0
   3227        0 TMP,    0
   3230        0 WORD,   0
   3231        0 BCNT,   0
 // SUBROUTINE PKR(FILNAM,NF)
/ REVISION -- JANUARY 22,1981
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- PACK USER GIVEN CHARACTERS INTO A 2-WORD
/               "FILE-NAME". THE 2 MOST SIGNIFICANT BITS
/                OF THE FILENAME ARE 00.
/ ARGUMENTS --
/       FILNAM - ADDRESS OF THE FIRST WORD OF THE FILENAME
/       NF - (RETURNED) THE NO. OF CHARACTERS IN THE FILENAME.
*3240
   3240        0 PKR,    0
```

```
--------
  //  FUNCTION PRTOCT(X)
   3241 3111240           MEMA    @PKR
   3242 2405301           ACCM    ADDR     /STORE THE ADDRESS OF THE FILENAME
   3243 3165301           ZERM    @ADDR
   3244 2135240           MPOAM   PKR
   3245 2405302           ACCM    NF       /STORE ADDRESS FOR RETURNING NF
   3246 3165302           ZERM    @NF
   3247  110006           MEMA    (6
   3250 2405064           ACCM    COUNT
   3251 2165303           ZERM    INDIC    /SET CR INDICATOR
   3252  462003 PKR1,     A-MZ    (3
   3253    1256           JMP     #+3
   3254 2125301           MPOM    ADDR
   3255 3165301           ZERM    @ADDR
   3256 2103303           MEMZ    INDIC    /IF INDIC HAS BEEN SET THEN JUST SH
   3257    1271           JMP     PKR2
   3260 3001304           JMS     @ECHO
   3261  462215           A-MZ    (215
   3262    1266           JMP     #+4
   3263 2025303           ONEM    INDIC
   3264 2165067           ZERM    TEMP
   3265    1271           JMP     PKR2
   3266 3125302           MPOM    @NF
   3267  470240           A-MA    (240
   3270 2405067           ACCM    TEMP
   3271 3111301 PKR2,     MEMA    @ADDR
   3272    5046           LLSH    6
   3273 2511067           A+MA    TEMP
   3274 3405301           ACCM    @ADDR
   3275 2717064           MMOMAZ  COUNT
   3276    1252           JMP     PKR1
   3277 2125240           MPOM    PKR      /INCREMENT FOR RETURN
   3300 1001240           JMP     @PKR
   3301       0 ADDR      0
   3302       0 NF,       0
   3303       0 INDIC,    0
/
/ EXTERNALS
   3304    2257 ECHO,     2257
--------
  //  FUNCTION PRTOCT(X)
ADDR      3301     BCNT      3231     BCOUNT    3070     COUNT     3064
COUNT1    3223     CRLF      2736     ECHO      3304     END       2720
INDIC     3303     INDIX     2730     L1        2671     L3        2706
LOOP      2661     MASK      3073     NBYTES    3226     NC        2727
NCC       2726     NF        3302     PAKF      3074     PAKF1     3121
PAKF2     3131     PAKF22    3146     PAKF3     3152     PAKF4     3160
PAKF5     3174     PAKF6     3202     PAKFE     3210     PAKFT     3215
```

```
PAKFLG    3225    PAKFX     3117    PKR       3240    PKR1      3252
PKR2      3271    POINT     3066    POINT1    3224    PRTO1     2615
PRTO2     2627    PRTOCT    2605    RECVF     3222    SENDF     3063
TEMP      3067    TEMP1     3071    TEMP2     3072    TMP       3227
TYPE      2731    UNP       2650    UNPF      2750    UNPF1     2775
UNPF2     3011    UNPF21    3026    UNPF22    3021    UNPF3     3031
UNPF33    3033    UNPF4     3043    UNPF5     3060    UNPFE     3061
UNPFLG    3065    UNPFX     2773    UNPFY     3005    UNPFZ     3014
WORD      3230
$
--------
  //   FUNCTION PRTOCT(X)
*
```

```
#
--------
/// SUBROUTINE DEC
 // SUBROUTINE DEC
/ REVISION -- DECEMBER  30,1980
/ AUTHOR   --BARRETT,TB
/ PURPOSE  -- TRANSFORM A PACKED ASCII HEX STRING TO BINARY AND PACK
/       5 NIBBLES PER NIC WORD
/ ARGUMENTS -- ALL ARGUMENTS ARE PASSED THROUGH A COMMON AREA
/       WITH THE FOLLOWING VARIABLES IN THE ORDER SHOWN-
/       WPNT - POINTS TO STORAGE LOCATION OF THE LAST WORD
/                STORED. THIS IS INCREMENTED WHENEVER A WORD IS
/                COMPLETE SO IT SHOULD BE SET ACCORDINGLY
/                ON INITIAL ENTRY.
/       WORD - CONTAINS THE NIBBLES OR PORTIONS THEREOF TO
/                BE STORED AT WPNT+1. IT SHOULD BE SET TO 0 ON
/                INITIAL CALL.
/       WCNT - A COUNTER FOR WORD. WHEN WORD IS EMPTY,WCNT
/                =5, WHEN FULL WCNT=0. WHEN WCNT GOES TO 0, WORD IS
/                STORED AT WPNT+1 AND WPNT IS INCREMENTED. SET TO 6
/                AT INITIAL CALL TO DEC.
/       COUNT - THE NUMBER OF NIBBLES+1 TO BE PACKED. NOTE THAT COUNT
/                IS DECREMENTED TO 0 BY DEC.
/       BCNT - BYTE COUNTER. SET TO 1 FOR INITIAL CALL.
/       APNT - POINT TO CURRENT STRING WORD. SET TO 1 LESS THAN THE
/                START OF THE STRING INITIALLY.
/       CHKSUM - NIBBLE VALUES ARE ADDED & STORED IN CHKSUM. SET TO
/                0 ON EACH CALL TO DEC (USUALLY).
/       NWORD - NO. OF WORDS STORED. (USUALLY SET TO 0 AT EACH CALL.
/       NOTE----
/       IN ORDER TO FORCE A WORD OUT OF DEC, SET WCNT TO COUNT.
/
////////////////////////////////////////////////////////////////////////////
*3310
   3310       0 DEC,    0
   3311 2707420 START,  MMOMZ   BCNT
   3312    1321         JMP     #+7
   3313  110005         MEMA    (5
   3314 2405420         ACCM    BCNT
   3315 2125421         MPOM    APNT
   3316 3111421         MEMA    @APNT
   3317    5001         LASH    1
   3320 2405406         ACCM    TEMP
   3321 2707416         MMOMZ   WCNT
   3322    1332         JMP     #+10
   3323  110005         MEMA    (5
   3324 2405416         ACCM    WCNT
   3325 2125414         MPOM    WPNT
```

```
    3326 2111415          MEMA    WORD
    3327 3405414          ACCM    @WPNT
    3330 2125423          MPOM    NWORD
    3331 2165415          ZERM    WORD
    3332 2707417          MMOMZ   COUNT
    3333    1337          JMP     #+4
    3334 2125420          MPOM    BCNT
    3335 2125416          MPOM    WCNT
    3336 1001310          JMP     @DEC     /NOTE EXIT
--------
/// SUBROUTINE DEC
    3337 2111420          MEMA    BCNT
    3340  462003          A-MZ    (3
    3341    1350          JMP     DEC1
    3342 2125421          MPOM    APNT
    3343 3111421          MEMA    @APNT
    3344 2405410          ACCM    TEMP1
    3345  405023          RISH    3
    3346 2511406          A+MA    TEMP
    3347    1357          JMP     DEC2
    3350  462002 DEC1,    A-MZ    (2
    3351    1356          JMP     DEC3
    3352 2111410          MEMA    TEMP1
    3353    5005          LASH    5
    3354 2405406          ACCM    TEMP
    3355    1357          JMP     DEC2
    3356 2111406 DEC3,    MEMA    TEMP
    3357    5144 DEC2,    EXCT    AC19
    3360 2511412          A+MA    TRAN
    3361    5003          LASH    3
    3362 2405406          ACCM    TEMP
    3363 2011413          ANDA    MASK
    3364 2405407          ACCM    NIBBLE
    3365    5044          LLSH    4
    3366 2505422          A+MM    CHKSUM
/ GET READY FOR NEXT WORD
    3367 2111406          MEMA    TEMP
    3370    5005          LASH    5
    3371 2405406          ACCM    TEMP
    3372  110006          MEMA    (6
    3373 2405411          ACCM    SHIF1
    3374 2111416          MEMA    WCNT
    3375 2325411          M-AM    SHIF1
    3376 2111407          MEMA    NIBBLE
    3377 2707411          MMOMZ   SHIF1
    3400 2162000          ZERZ
    3401   01404          JMP     #+3
    3402  405024          RISH    4
    3403    1377          JMP     #-4
```

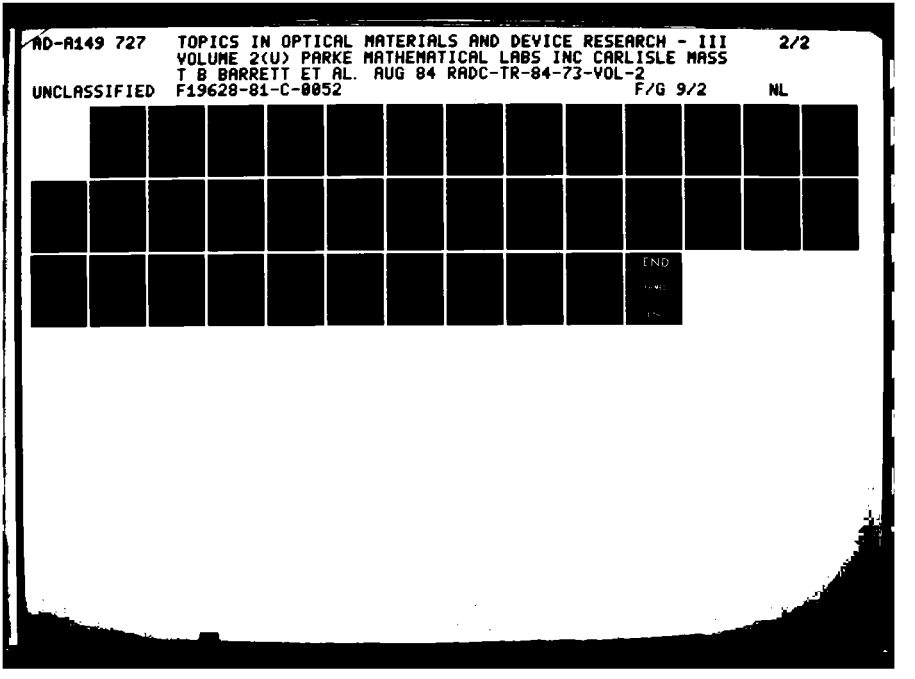
```
   3404 2505415          A+MM     WORD
   3405    1311          JMP      START
/ SCRATCH STORAGE
   3406       0 TEMP,    0
   3407       0 NIBBLE,  0
   3410       0 TEMP1,   0
   3411       0 SHIF1,   0
/ MASKS
   3412  220000 TRAN,    220000
   3413 3600000 MASK,    3600000
/COMMON STORAGE
   3414       0 WPNT,    0
   3415       0 WORD,    0
   3416       0 WCNT,    0
   3417       0 COUNT,   0
   3420       0 BCNT,    0
   3421       0 APNT,    0
--------
/// SUBROUTINE DEC
   3422       0 CHKSUM,  0
   3423       0 NWORD,   0
 // SUBROUTINE NIBBIN(VALUE,NNIB,C,BIN)
/ REVISION -- DECEMBER 30,1980
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- CONVERT PACKED BCD OR BCH TO BINARY.
/ ARGUMENTS --
/       VALUE - PACKED BCD OR BCH; MOST SIGNIFICANT NIBBLE
/               AT UPPER ORDER LOCATION IN VALUE;LEFT JUSTIFIED.
/       NNIB  - NO.OF 4 BIT NIBBLES TO BE CONVERTED
/       C     - OCT 12 IF THIS IS A BCD VALUE
/               OCT 20 IF THIS IS A BCH VALUE
/       BIN   - RETURNS BINARY VALUE HERE
/
////////////////////////////////////////////////////////////////////////
*3470
   3470       0 NIBBIN,  0
   3471  110006          MEMA     (6
   3472 2405560          ACCM     SHIF
   3473 3111470          MEMA     @NIBBIN
   3474 2405553          ACCM     VALUE
   3475 2125470          MPOM     NIBBIN
   3476 2165557          ZERM     RESULT
   3477 3113470          MEMAZ    @NIBBIN
   3500 2162000          ZERZ
   3501    1550          JMP      NIB2
   3502 2405554          ACCM     NNIB
   3503 2325560          M-AM     SHIF
   3504 2111553          MEMA     VALUE
   3505 2707560          MMOMZ    SHIF
```

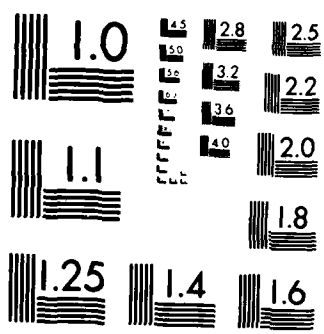
```
    3506 2162000           ZERZ
    3507    1512           JMP      #+3
    3510  405024           RISH     4
    3511    1505           JMP      #-4
    3512 2405553           ACCM     VALUE
    3513 2125470           MPOM     NIBBIN
    3514 3111470           MEMA     @NIBBIN
    3515 2405555           ACCM     C
    3516 2135470           MPOAM    NIBBIN
    3517 2405556           ACCM     BIN
    3520 2125470           MPOM     NIBBIN
    3521 2111553           MEMA     VALUE
    3522 2025526           ONEM     MPLCND
    3523   10017 NIB1,     ANDA     (17
    3524    4354           TACMQ
    3525  505320           MULT
    3526       0 MPLCND,    0
    3527    4343           TMQAC
    3530 2505557           A+MM     RESULT
    3531 2707554           MMOMZ    NNIB
    3532    1534           JMP      #+2
    3533    1550           JMP      NIB2
    3534 2111526           MEMA     MPLCND
    3535 2405541           ACCM     #+4
    3536 2111555           MEMA     C
--------
/// SUBROUTINE DEC
    3537    4354           TACMQ
    3540  505320           MULT
    3541       0           0
    3542    4343           TMQAC
    3543 2405526           ACCM     MPLCND
    3544 2111553           MEMA     VALUE
    3545  405024           RISH     4
    3546 2405553           ACCM     VALUE
    3547    1523           JMP      NIB1
    3550 2111557 NIB2,     MEMA     RESULT
    3551 3405556           ACCM     @BIN
    3552 1001470           JMP      @NIBBIN
    3553       0 VALUE,     0
    3554       0 NNIB,      0
    3555       0 C,         0
    3556       0 BIN,       0
    3557       0 RESULT,    0
    3560       0 SHIF,      0
 // SUBROUTINE TEKHEX(CNT,NBYTE)
/ REVISION -- JANUARY 24,1981
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- DECODES (TO BINARY) A TEKTRONIX HEX  ILE AND PUTS
```

```
/                THE RESULT IN PACKED FORM STARTING AT LOCATION
/                100000. THE INPUT FILE IS ASSUMED TO BE IN
/                PACKED FORM STARTING AT 100000.
/ ARGUMENTS -- CNT - TOTAL NO. OF PACKED WORDS IN THE OUTPUT.
/                NOTE THAT THE LAST PACKED WORD MAY HAVE 0 FILL.
/                ALSO NOTE THAT DEC PACKS WORDS SUCH THAT NIBBLE1
/                OCCUPIES 19-16....,NIBBLE5, 3-0.
/                NBYTE - TOTAL NO. OF BYTES (2 NIBS) STORED.
/       - NOTE THAT THIS SUBROUTINE ASSUMES THAT THE TEK. 8002 AND
/       COMM-STOR ARE CONFIGURED SUCH THAT EACH "LEADING" SLASH IS
/       PREFACED WITH XOF (ASCII 223). ALSO NOTE THAT THERE ARE
/       17 (21 OCT) SURPLUS CHARS. AFTER THE LAST CR.
/       BE SURE TO USE MX MODE WHEN STORING THE HEX FILE.
*3565
   3565        0 TEKHEX, 0
   3566 2111565          MEMA    TEKHEX
   3567 2405750          ACCM    CNT
   3570 3165750          ZERM    @CNT
   3571 2135565          MPOMA   TEKHEX
   3572 2405754          ACCM    NBYTE
   3573 3165754          ZERM    @NBYTE
   3574 2125565          MPOM    TEKHEX
   3575 2025420          ONEM    BCNT     /INITIALIZE DEC
   3576 2165746          ZERM    WORDO
   3577  110006          MEMA    (6
   3600 2405747          ACCM    WCNTO
   3601 2111756          MEMA    BUFS
   3602 2550000          AMOA
   3603 2405421          ACCM    APNT
   3604 2405745          ACCM    WPNTO
   3605  110013 TEK1,    MEMA    (13
   3606 2405417          ACCM    COUNT
   3607  110006          MEMA    (6
--------
/// SUBROUTINE DEC
   3610 2405416          ACCM    WCNT
   3611 2165415          ZERM    WORD
   3612 2111755          MEMA    HEADA    /ADDRESS OF HEADER START
   3613 2405414          ACCM    WPNT
   3614 2001310          JMS     DEC
   3615 2111751          MEMA    HEAD     /CHECK FOR "/"
   3616    5050          LLSH    10
   3617   10017          ANDA    (17
   3620  462017          A-MZ    (17
   3621 2162000          ZERZ
   3622    1634          JMP     TEK2
   3623 3001760          JMS     @CRLF
   3624 3001757          JMS     @UNP
   3625      12          12
```

```
3626        1           1
3627     6445           TEXT % TE
3630   535045   KHE
3631   700045   X E
3632   627700   R%
3633     1744           JMP     TEKE     / GOTO TEKHEX END
3634  2111752 TEK2,     MEMA    HEAD+1
3635     5044           LLSH    4
3636  2405640           ACCM    #+2
3637  2001470           JMS     NIBBIN
3640        0           0
3641        2           2
3642       20           20
3643        0           0
3644  2111643           MEMA    #-1
3645  3505754           A+MM    @NBYTE
3646     5001           LASH    1        /*2 FOR TOTAL NO. OF NIBBLES
3647  2407417           ACCMZ   COUNT
3650     1652           JMP     #+2
3651     1734           JMP     TEK3     /NORMAL TERMINATION-CHECK FOR WHETHI
/         THIS WAS LAST FULL BYTE.
3652  2125417           MPOM    COUNT
3653  2111745           MEMA    WPNTO
3654  2405414           ACCM    WPNT
3655  2111746           MEMA    WORDO
3656  2405415           ACCM    WORD
3657  2111747           MEMA    WCNTO
3660  2405416           ACCM    WCNT
3661  2165422           ZERM    CHKSUM
3662  2165423           ZERM    NWORD
3663  2001310           JMS     DEC
3664  2111422           MEMA    CHKSUM
3665    10377           ANDA    (377     /THIS MOD 256
3666  2405753           ACCM    CHKS
3667  2111423           MEMA    NWORD
3670  3505750           A+MM    @CNT
3671  2111414           MEMA    WPNT     /STORE STATE FOR NEXT CALL
3672  2405745           ACCM    WPNTO
3673  2111415           MEMA    WORD
3674  2405746           ACCM    WORDO
3675  2111416           MEMA    WCNT
--------
/// SUBROUTINE DEC
3676  2405747           ACCM    WCNTO
3677   110004           MEMA    (4       /SET FOR GETTING TRAILER (3 NIBBLES
3700  2405417           ACCM    COUNT
3701   110004           MEMA    (4
3702  2405416           ACCM    WCNT
3703  2165415           ZERM    WORD
```

```
3626        1           1
3627     6445           TEXT % TE
3630   535045  KHE
3631   700045  X E
3632   627700  R%
3633     1744           JMP     TEKE     / GOTO TEKHEX END
3634 2111752 TEK2,      MEMA    HEAD+1
3635     5044           LLSH    4
3636 2405640           ACCM    #+2
3637 2001470           JMS     NIBBIN
3640        0           0
3641        2           2
3642       20           20
3643        0           0
3644 2111643           MEMA    #-1
3645 3505754           A+MM    @NBYTE
3646     5001           LASH    1        /*2 FOR TOTAL NO. OF NIBBLES
3647 2407417           ACCMZ   COUNT
3650     1652           JMP     #+2
3651     1734           JMP     TEK3     /NORMAL TERMINATION-CHECK FOR WHETHI
/       THIS WAS LAST FULL BYTE.
3652 2125417           MPOM    COUNT
3653 2111745           MEMA    WPNTO
3654 2405414           ACCM    WPNT
3655 2111746           MEMA    WORDO
3656 2405415           ACCM    WORD
3657 2111747           MEMA    WCNTO
3660 2405416           ACCM    WCNT
3661 2165422           ZERM    CHKSUM
3662 2165423           ZERM    NWORD
3663 2001310           JMS     DEC
3664 2111422           MEMA    CHKSUM
3665    10377          ANDA    (377     /THIS MOD 256
3666 2405753           ACCM    CHKS
3667 2111423           MEMA    NWORD
3670 3505750           A+MM    @CNT
3671 2111414           MEMA    WPNT     /STORE STATE FOR NEXT CALL
3672 2405745           ACCM    WPNTO
3673 2111415           MEMA    WORD
3674 2405746           ACCM    WORDO
3675 2111416           MEMA    WCNT
--------
/// SUBROUTINE DEC
3676 2405747           ACCM    WCNTO
3677    110004         MEMA    (4       /SET FOR GETTING TRAILER (3 NIBBLES
3700 2405417           ACCM    COUNT
3701    110004         MEMA    (4
3702 2405416           ACCM    WCNT
3703 2165415           ZERM    WORD
```


AD-A149 727
TOPICS IN OPTICAL MATERIALS AND DEVICE RESEARCH - III
VOLUME 2(U) PARKE MATHEMATICAL LABS INC CARLISLE MASS
T B BARRETT ET AL. AUG 84 RADC-TR-84-73-VOL-2
2/2
UNCLASSIFIED
F19628-81-C-0052
F/G 9/2
NL
END



```
/// SUBROUTINE DEC
APNT        3421      BCNT        3420      BIN         3556      BUFS        3756
C           3555      CHKS        3753      CHKSUM      3422      CNT         3750
COUNT       3417      CRLF        3760      DEC         3310      DEC1        3350
DEC2        3357      DEC3        3356      HEAD        3751      HEADA       3755
MASK        3413      MPLCND      3526      NBYTE       3754      NIB1        3523
NIB2        3550      NIBBIN      3470      NIBBLE      3407      NNIB        3554
NWORD       3423      RESULT      3557      SHIF        3560      SHIF1       3411
START       3311      TEK1        3605      TEK2        3634      TEK3        3734
TEKE        3744      TEKHEX      3565      TEMP        3406      TEMP1       3410
TRAN        3412      UNP         3757      VALUE       3553      WCNT        3416
WCNTO       3747      WORD        3415      WORDO       3746      WPNT        3414
WPNTO       3745
$
--------
/// SUBROUTINE DEC
W
*
```

```
#
--------
// SUBROUTINE NICFIL(NW,FILNAM)
 // SUBROUTINE NICFIL(NW,FILNAM)
/ REVISION -- JANUARY 26,1980
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- TRANSFER A CORE "FILE" TO DISK FILE. THE CORE FILE
/               STARTS AT 100000.
/ ARGUMENTS --
/       NW - LENGTH OF THE FILE (WORDS)
/       FILNAM - ADDRESS OF THE NAME TO BE ASSIGNED TO THE FILE.
/
////////////////////////////////////////////////////////////////
*4010
   4010        0 NICFIL, 0
   4011  3110010         MEMA    @NICFIL
   4012  3404425         ACCM    @OARG2
   4013  2404032         ACCM    NW
   4014  3164424         ZERM    @OARG1
   4015  2124010         MPOM    NICFIL
   4016  3110010         MEMA    @NICFIL
   4017  2404063         ACCM    AFIL
   4020  2124010         MPOM    NICFIL  /SET FOR ERROR RETURN
   4021  3000420         JMS     @OPENW
   4022  3110425         MEMA    @OARG2  /MAKE SURE HAVE ENOUGH SPACE
   4023  2462032         A-MZ    NW
   4024     5144         EXCT    AC19    /IF NEG. JUMP TO ERROR EXIT
   4025       47         JMP     ERR
   4026  3110424         MEMA    @OARG1
   4027  2404031         ACCM    IT
   4030  3000421         JMS     @WRITE
   4031        0 IT,     0
   4032        0 NW,     0
   4033   100000         100000
   4034  2110032         MEMA    NW
   4035  3404425         ACCM    @OARG2
   4036  3110063         MEMA    @AFIL
   4037  2404044         ACCM    #+5
   4040  2124063         MPOM    AFIL
   4041  3110063         MEMA    @AFIL
   4042  2404045         ACCM    #+3
   4043  3000422         JMS     @CLOSE
   4044        0         0
   4045        0         0
   4046       61         JMP     NIC1
   4047  3000412 ERR,    JMS     @UNP
   4050       17         17
   4051        1         1
```

```
   4052    5657          TEXT % NO
   4053    4651   FI
   4054  544500  LE
   4055  636041  SPA
   4056  434515  CE-
   4057  770000  %
   4060 1000010          JMP     @NICFIL
   4061 2124010 NIC1,    MPOM    NICFIL
--------
// SUBROUTINE NICFIL(NW,FILNAM)
   4062 1000010          JMP     @NICFIL
/ SCRATCH STORAGE
   4063       0 AFIL,    0
 // SUBROUTINE SEARCH(MNEM,CODE,VALUE,NBYTE,TABLA,FLAG)
/ REVISION -- JANUARY 22,1981
/ AUTHOR   -- BARRETT,TB
/ PURPOSE  -- SEARCH A TABLE FOR CODE&VALUE&NO. OF BYTES IN THE
/               VALUE (WHERE VALUE IS A SINGLE WORD WITH UP TO
/               5 NIBBLES) OR INSERT VALUE & NO. OF NIBBLES IN VALUE
/ PARAMETERS --
/       MNEM - 3 LETTER MNEMONIC (PACKED) WHICH IDENTIFIES AN ENTRY
/       CODE - THE CORRESPONDING CODE (CAMERA "SECONDARY ADDRESS")
/       VALUE - DATA WORD ASSOCIATED WITH THE CODE (MAY BE 0)
/               TYPICALLY THIS IS A BCD CODE. FOR EXAMPLE 1203
/               WOULD HAVE NIBBLES 1,2,0,3,0 IN THAT ORDER FOR
/               AN OCTAL WORD = 0220060.
/       NBYTE - NO. OF NIBBLES IN VALUE (MAY BE 0)
/       TABLA - ADDRESS OF THE START OF THE TABLE
/       FLAG - ON ENTRY, FLAG IS USED TO INDICATE WHETHER THIS
/               IS A RETURN (0) OR REPLACE (1) OPERATION. ON
/               SEARCH OPERATIONS, FLAG IS ALSO RETURNED AS 0 FOR
/               A SUCCESSFUL SEARCH AND AS 1 FOR NO-FIND.
/       NOTE ---
/       TABLE HAS THE FORM -
/       MNEMONIC (3 PACKED LETTERS,R-JUSTIFIED)
/       CODE  RIGHT JUSTIFIED 8-BIT CODE
/       N-NIBBLES NO. OF NIBBLES IN VALUE
/       VALUE PACKED NIBBLES
/       ETC.
/       THE TABLE SHOULD BE TERMINATED WITH @.
*4070
   4070       0 SEARCH,  0
   4071 3110070          MEMA    @SEARCH
   4072 2404150          ACCM    MNEM
   4073 2134070          MPOMA   SEARCH
   4074 2404151          ACCM    CODE
   4075 2134070          MPOMA   SEARCH
   4076 2404304          ACCM    VALUE
   4077 2134070          MPOMA   SEARCH
```

```
   4100 2404152           ACCM    NBYTE
   4101 2124070           MPOM    SEARCH
   4102 3110070           MEMA    @SEARCH
   4103 2404153           ACCM    TABLA
   4104 2124070           MPOM    SEARCH
   4105      112          JMP     #+5
   4106 2124153 SEA1,     MPOM    TABLA
   4107 2124153           MPOM    TABLA
   4110 2124153           MPOM    TABLA
   4111 2124153           MPOM    TABLA
   4112 3110153           MEMA    @TABLA
   4113  462300           A-MZ    (300
   4114      116          JMP     #+2
   4115      145          JMP     SEA2    /CANT FIND
   4116 2462150           A-MZ    MNEM
   4117      106          JMP     SEA1
--------
// SUBROUTINE NICFIL(NW,FILNAM)
   4120 3102070           MEMZ    @SEARCH
   4121      134          JMP     SEA3    /STORE
   4122 2124153           MPOM    TABLA
   4123 3110153           MEMA    @TABLA
   4124 3404151           ACCM    @CODE
   4125 2124153           MPOM    TABLA
   4126 3110153           MEMA    @TABLA
   4127 3404152           ACCM    @NBYTE
   4130 2124153           MPOM    TABLA
   4131 3110153           MEMA    @TABLA
   4132 3404304           ACCM    @VALUE
   4133      144          JMP     SEA4
   4134 2124153 SEA3,     MPOM    TABLA
   4135 2124153           MPOM    TABLA
   4136 3110152           MEMA    @NBYTE
   4137 3404153           ACCM    @TABLA
   4140 2124153           MPOM    TABLA
   4141 3110304           MEMA    @VALUE
   4142 3404153           ACCM    @TABLA
   4143      145          JMP     SEA2
   4144 3166070 SEA4,     ZERMZ   @SEARCH
   4145 3024070 SEA2,     ONEM    @SEARCH
   4146 2124070           MPOM    SEARCH
   4147 1000070           JMP     @SEARCH
 / SCRATCH STORAGE
   4150        0 MNEM,    0
   4151        0 CODE,    0
   4152        0 NBYTE,   0
   4153        0 TABLA,   0
  // SUBROUTINE ZERTAB(TABLE)
 / PURPOSE  -- ZEROES THE COMMAND TABLE
```

```
/ ARGUMENTS -- TABLE - START ADDRESS OF TABLE
*4160
   4160        0 ZERTAB, 0
   4161   110023          MEMA    (23
   4162  2404173          ACCM    COUNT
   4163  3110160          MEMA    @ZERTAB
   4164  2404174          ACCM    POINT
   4165  3164174 ZE1,     ZERM    @POINT
   4166  2124174          MPOM    POINT
   4167  2706173          MMOMZ   COUNT
   4170      165          JMP     ZE1
   4171  2124160          MPOM    ZERTAB
   4172  1000160          JMP     @ZERTAB
/ SCRATCH
   4173        0 COUNT,  0
   4174        0 POINT,  0
/
////////////////////////////////////////////////////////////////
/
// FUNCTION MULTP(X)
/ PURPOSE  -- MULTIPLIES ACC BY X AND RETURNS RESULT IN ACC (LOWER
/             20 BITS OF THE RESULT. THE HIGH ORDER BITS ARE PUT
/               IN THE MQ REGISTER.
   4175        0 MULTP,  0
--------
// SUBROUTINE NICFIL(NW,FILNAM)
   4176     4354          TACMQ   /TRANSFER ACC TO MQ REGISTER
   4177  3110175          MEMA    @MULTP
   4200  2404202          ACCM    #+2
   4201   505320          MULT
   4202        0          0
   4203  2404304          ACCM    VALUE
   4204     4343          TMQAC   /TRANSFERS MQ (LOW ORDER) TO ACC
   4205  2404305          ACCM    VALUE1
   4206  2110304          MEMA    VALUE
   4207     4354          TACMQ   /PUT HIGH ORDER IN MQ
   4210  2110305          MEMA    VALUE1  /LOW ORDER IN ACC FOR RETURN
   4211  2124175          MPOM    MULTP   /SET FOR RETURN
   4212  1000175          JMP     @MULTP
/
////////////////////////////////////////////////////////////////
/
// FUNCTION DIVDE(X)
/ PURPOSE -- DIVIDE MQ+ACC BY X AND RETURN THE RESULT IN ACC.
/       RETURN THE REMAINDER IN X. NOTE THAT MQ CONTAINS THE HIGH
/       ORDER BITS AND ACC THE LOW ORDER BITS OF THE DIVIDEND.
   4213        0 DIVDE,  0
   4214     5210          CLL     /CLEAR LINK
   4215     5144          EXCT    AC19
```

```
   4216    5204          STL
   4217    5001          LASH    1        /LEFT SHIFT DIVIDEND
   4220 2404304          ACCM    VALUE    /STORE TEMPORARILY
   4221    4343          TMQAC   /GET HIGH ORDER BITS
   4222    5001          LASH    1
   4223    5141          EXCT    L
   4224 2430000          APOA
   4225 2404305          ACCM    VALUE1   /STORE TEMPORARILY
   4226 2110304          MEMA    VALUE
   4227    4354          TACMQ   /LOAD IT INTO MQ
   4230 3110213          MEMA    @DIVDE   /GET DIVISOR
   4231 2404234          ACCM    D1
   4232 2110305          MEMA    VALUE1   /PUT HIGH ORDER IN ACC
   4233  465300          DIVD
   4234       0 D1,      0
   4235  405021          RISH    1        /RESTORE THE REMAINDER
   4236 3404213          ACCM    @DIVDE   /AND STORE FOR RETURN
   4237    4343          TMQAC   /QUOTIENT TO ACC
   4240 2124213          MPOM    DIVDE    /FOR RETURN
   4241 1000213          JMP     @DIVDE
/
////////////////////////////////////////////////////////////////////////////}
/
// SUBROUTINE SENDF
/ PURPOSE -- GIVEN A "WORD" TO BE PRINTED IN ACC, SENDF PRINTS THE
/       WORD ACCORDING TO THE FOLLOWING RULE-
/ (1) IF SMODE=0 THEN PRINT DIRECTLY AFTER CONVERTING TO
/       NIC ASCII
/ (2) IF SMODE=1 THEN CONVERT TO DOUBLE HEX AND PRINT. ALSO COUNT
/       CHARACTERS (OR WORDS) AND AT END OF WCNTO WORDS DO A CR. THE
/       WORD COUNTER COUNTER SHOULD BE SET TO WCNTO INITIALLY.
/ SMODE,WCNT  AND WCNTO CAN BE CONSIDERED IN COMMON DECLARED IN SENDF.
--------
// SUBROUTINE NICFIL(NW,FILNAM)
   4242       0 SENDF,  0
   4243 2102311         MEMZ    SMODE
   4244     250         JMP     HEXM
   4245  510200         A+MA    (200
   4246 3000417         JMS     @TYPE
   4247     276         JMP     HEXM2
   4250 2024307 HEXM,   ONEM    LCNT     /SET NIBBLE COUNT TO 2
   4251 2124307         MPOM    LCNT
   4252    5064         RLSH    4
   4253 2404304 HEXM1,  ACCM    VALUE
   4254  110260         MEMA    (260
   4255 2404306         ACCM    PREF
   4256 2110304         MEMA    VALUE
   4257   10017         ANDA    (17
   4260 2404305         ACCM    VALUE1
```

```
   4261  470012          A-MA    (12
   4262    5144          EXCT    AC19
   4263     267          JMP     #+4
   4264 2424305          APOM    VALUE1
   4265  110300          MEMA    (300
   4266 2404306          ACCM    PREF
   4267 2110305          MEMA    VALUE1
   4270 2510306          A+MA    PREF
   4271 3000417          JMS     @TYPE
   4272 2110304          MEMA    VALUE
   4273    5044          LLSH    4
   4274 2706307          MMOMZ   LCNT
   4275     253          JMP     HEXM1
   4276 2706310 HEXM2,   MMOMZ   WCNT
   4277     303          JMP     SENDE
   4300 2110312          MEMA    WCNTO
   4301 2404310          ACCM    WCNT
   4302 3000414          JMS     @CRLF
   4303 1000242 SENDE,   JMP     @SENDF
/ SCRATCH STORAGE
   4304       0 VALUE,   0
   4305       0 VALUE1,  0
   4306       0 PREF,    0
   4307       0 LCNT,    0
/ COMMON
   4310       0 WCNT,    0
   4311       0 SMODE,   0
   4312       0 WCNTO,   0
/
////////////////////////////////////////////////////////////////////}
/
// SUBROUTINE GETFIL
/ PURPOSE -- ASKS FOR FILE WANTED AND GETS THE FILE INTO NIC
/       CORE STARTING AT 100000. IF THE FILE IS UNAVAILABLE,
/              AN ERROR MESSAGE IS PRINTED AND THE
/       ERROR EXIT IS TAKEN. (NORMAL EXIT IS 2 BEYOND CALL POINT.)
   4313       0 GETFIL, 0
   4314 3000412          JMS     @UNP
   4315      14          14
   4316       0          0
--------
// SUBROUTINE NICFIL(NW,FILNAM)
   4317    4651          TEXT % FI
   4320  544500  LE
   4321  564155  NAM
   4322  453735  E?=
   4323  770000  %
   4324 3000413          JMS     @PKR
   4325    4331          FILNAM
```

```
   4326        0           0
   4327  3000414           JMS     @CRLF
   4330  3000415           JMS     @OPENR
   4331        0 FILNAM,   BLOCK   2
   4333  3110424           MEMA    @OARG1
   4334     5104           SKIP    AC19
   4335      351           JMP     GETF1
   4336  3000412           JMS     @UNP    /FILE DOES NOT EXIST
   4337       24           24
   4340        1           1
   4341     4651           TEXT % FI
   4342   544500  LE
   4343   445745  DOE
   4344   630056  S N
   4345   576400  OT
   4346   457051  EXI
   4347   636477  ST%
   4350  1000313           JMP     @GETFIL
   4351  3110424 GETF1,    MEMA    @OARG1
   4352  2404356           ACCM    ITG
   4353  3110425           MEMA    @OARG2
   4354  2404357           ACCM    SIZE
   4355  3000416           JMS     @READD
   4356        0 ITG,      0
   4357        0 SIZE,     0
   4360  2124313           MPOM    GETFIL
   4361  1000313           JMP     @GETFIL
/
/ SUBROUTINE PAUSE
/ PURPOSE -- VARIABLE PAUSE
   4362        0 PAUSE,    0
   4363  3110362           MEMA    @PAUSE
   4364  2404173           ACCM    COUNT
   4365  2706173           MMOMZ   COUNT
   4366      365           JMP     #-1
   4367  2124362           MPOM    PAUSE
   4370  1000362           JMP     @PAUSE
/
/ SUBROUTINE OCT(X)
/ PURPOSE -- PACKS USER GIVEN OCTAL VALUE (UP TO 7 DIGITS) INTO LOCATION
/ X. THE VALUE IS RIGHT JUSTIFIED,ZERO FILL (E.G.1 => 0000001).
   4371        0 OCT,      0
   4372  3110371           MEMA    @OCT
   4373  2404304           ACCM    VALUE
   4374  3164304           ZERM    @VALUE  /ZERO FILL THE NUMBER
   4375  3000423 OCT1,     JMS     @ECHO   /GET NEXT DIGIT
   4376   462215           A-MZ    (215    /IF CR WE ARE THROUGH
   4377  2162000           ZERZ
--------
```

```
// SUBROUTINE NICFIL(NW,FILNAM)
   4400     410          JMP     OCTE
   4401  470260          A-MA    (260     /CONVERT FROM ASCII TO OCTAL
   4402 2404305          ACCM    VALUE1
   4403 3110304          MEMA    @VALUE
   4404    5043          LLSH    3
   4405 2510305          A+MA    VALUE1
   4406 3404304          ACCM    @VALUE   /STORE NEW VALUE
   4407     375          JMP     OCT1
   4410 2124371 OCTE,    MPOM    OCT
   4411 1000371          JMP     @OCT
/
/ EXTERNALS
   4412    2650 UNP,     2650
   4413    3240 PKR,     3240
   4414    2736 CRLF,    2736
   4415    2432 OPENR,   2432
   4416    2511 READD,   2511
   4417    2731 TYPE,    2731
   4420    2420 OPENW,   2420
   4421    2470 WRITE,   2470
   4422    2451 CLOSE,   2451
   4423    2257 ECHO,    2257
/ DEFINITIONS
   4424    7770 OARG1,   7770
   4425    7771 OARG2,   7771
/
//////////////////////////////////////
/
--------
// SUBROUTINE NICFIL(NW,FILNAM)
AFIL      4063     CLOSE     4422     CODE      4151     COUNT     4173
CRLF      4414     D1        4234     DIVDE     4213     ECHO      4423
ERR       4047     FILNAM    4331     GETF1     4351     GETFIL    4313
HEXM      4250     HEXM1     4253     HEXM2     4276     IT        4031
ITG       4356     LCNT      4307     MNEM      4150     MULTP     4175
NBYTE     4152     NIC1      4061     NICFIL    4010     NW        4032
OARG1     4424     OARG2     4425     OCT       4371     OCT1      4375
OCTE      4410     OPENR     4415     OPENW     4420     PAUSE     4362
PKR       4413     POINT     4174     PREF      4306     READD     4416
SEA1      4106     SEA2      4145     SEA3      4134     SEA4      4144
SEARCH    4070     SENDE     4303     SENDF     4242     SIZE      4357
SMODE     4311     TABLA     4153     TYPE      4417     UNP       4412
VALUE     4304     VALUE1    4305     WCNT      4310     WCNTO     4312
WRITE     4421     ZE1       4165     ZERTAB    4160
$
--------
// SUBROUTINE NICFIL(NW,FILNAM)
W
```

APPENDIX F - CTLS Source Listing

```
Tektronix     Z80 ASM V3.3  NIC-488-CTL                  Page    1

00002                   ;OCTOBER 28,1981
00003
00004
00005                 LIST
00006                 NOLIST   MLB
00007                   GLOBAL  COM1,COM2,COM3,COM4,COM5,COM6
00008                   ;GPIB CONTROLLER SUBROUTINES ADAPTED FROM INTEL
00009                   ;PERIPHERAL DESIGN HANDBOOK, AUG. 80,P 2-218
00010             ;
00011             ;       MEMORY CONTROL
00012             ;
00013  0000       CSEG    EQU     00H
00014  0800       DSEG    EQU     800H
00015  0BFF       STKBOT  EQU     0BFFH
00016  0900       COM8    EQU     900H
00017             ;
00018             ;       INITIAL DATA
00019             ;
00020  0004       MXBLK   EQU     4       ;Max. no. of 256 byte blocks to be stored in the CTL buffer.
00021  0008       LASTC   EQU     8       ;last legitimate command number
00022             ;
00023             ;       8291 CONTROL VALUES
00024             ;
00025  0020       PRT91   EQU     20H     ;8291 Base Port #
00026             ;
00027             ;       Reg #0 Data-in &Data-out
00028  0020       DIN     EQU     PRT91+0 ;Data-in reg
00029  0020       DOUT    EQU     PRT91+0 ;Data-out reg
00030             ;
00031                     ;Reg #1 Interrupt 1 Constants
00032  0021       INT1    EQU     PRT91+1 ;INT Reg 1
00033  0001       BOM     EQU     1       ;BO status bit no.
00034  0000       BIM     EQU     0       ;BI status bit no.
00035  0010       ENDMK   EQU     10H     ;91 END INTERR Mask
00036  0080       CPT     EQU     80H     ;91 command pass through int bit
00037             ;
00038             ;
00039             ;       Reg #2 Interrupt 2
00040  0022       INT2    EQU     PRT91+2
00041             ;
00042             ;
00043             ;       Reg #4 Address Mode Constants
00044  0024       ADRMD   EQU     PRT91+4 ;91 address mode register #
00045  0080       TON     EQU     80H     ;91 talk only mode & not listen only
00046  0040       LON     EQU     40H     ;91 listen only and not ton
00047  0001       MODE1   EQU     01      ;91 mode 1 addressing
00048             ;
00049             ;       Reg #4 (read)
00050  0024       ADRST   EQU     PRT91+4
00051  0002       TA      EQU     2       ;Talk active
00052             ;
00053             ;
00054             ;       Reg #5 (write) Auxiliary Mode Register
```

```
00055  0025   AUXMD  EQU   PRT91+5 ;91 auxiliary mode register #
00056  0024   CLKRT  EQU   24h     ;91 4 Mhz clock input
00057  0002   INITIAL EQU   02      ;91 reset
00058  0003   FNHSK  EQU   03      ;91 finish handshake command
00059  000F   VSCMD  EQU   0FH     ;91 Valid command pass-through
00060  0006   SEOI   EQU   06H     ;91 send EOI
00061  0080   AXRA   EQU   80H     ;91aux. reg A pattern
00062  0002   HOEND  EQU   2       ;91 hold off handshake on end
00063  0008   EOIS   EQU   8       ;91 output EOI on EOS sent
00064  0004   EOEOS  EQU   4       ;91 end on EOS received
00065  00A0   AXRB   EQU   0A0H    ;Aux. reg. B pattern
00066  0001   CPTEN  EQU   01H     ;Command pass-through enable
00067                 ;
00068                 ;       Reg #5  (read)
00069  0025   CPTRG  EQU   PRT91+5 ;Command Pass-through  reg
00070                 ;
00071                 ;
00072                 ;       Reg #6  Address 0/1 reg. constants
00073  0026   ADRG1  EQU   PRT91+6
00074  0060   DTDL1  EQU   60H     ;Disable major talker & listener
00075  00E0   DTDL2  EQU   0E0H    ;Disable minor talker & listener
00076                 ;
00077                 ;
00078                 ;       Reg #7  EOS     Character Register
00079  0027   EOSR   EQU   PRT91+7
00080                 ;
00081                 ;
00082                 ;       8292    CONTROL VALUES
00083                 ;
00084                 ;
00085  0010   PRT92  EQU   10H     ;8298 Base Port #
00086                 ;
00087  0010   INTMR  EQU   PRT92+0 ;92 INTRP Mask Reg
00088  00A0   INTM   EQU   0A0H    ;TCI
00089                 ;
00090  0010   ERRM   EQU   PRT92+0 ;92 error mask register
00091                 ;
00092  0010   ERFLAG EQU   PRT92+0 ;error flag pseudo register
00093  0002   TOUT2  EQU   02      ;92 time for standby
00094  0004   TOUT3  EQU   04              ;92 time out for TCST
00095                 ;
00096  0010   TOREG  EQU   PRT92+0 ;92 time out pseudo-register
00097  007F   TMOUT  EQU   7FH     ;Time out byte for TOREG
00098                 ;
00099  0011   CMD92  EQU   PRT92+1 ;92 Command Register
00100  0011   INTST  EQU   PRT92+1 ;92 Interrupt Status Register
00101  0002   IBFBT  EQU   2       ;Input Buffer Full bit
00102  0005   SRQ    EQU   05      ;92 SRQ bit no.
00103  0020   SRQBT  EQU   20H     ;SRQ Bit
00104  0040   ERRBT  EQU   40H     ;ERR Bit
00105                 ;
00106  0010   CLRST  EQU   PRT92+0;92 Controller Status pseudo-reg
00107  0009   SYCS   EQU   09H     ;Control Switch status
```

```
00108  0040   CABT   EQU   40H     ;Controller active bit
00109         ;
00110  0010   TOST   EQU   PRT92+0 ;92 time out pseudo-register
00111         ;
00112  0010   BUSST  EQU   PRT92+0 ;92 GPIB status pseudo-register
00113  0008   SYCBT  EQU   08H     ;SYC status bit
00114         ;
00115         ;      8292   OPERATION COMMANDS
00116  00F1   GIDL   EQU   0F1H    ;Go to idle
00117  00F2   RSET   EQU   0F2H     ;Reset
00118  00F3   RSTI   EQU   0F3H    ;reset interrupts
00119  00F6   GTSB   EQU   0F6H    ;Goto standby
00120  00F9   ABORT  EQU   0F9H    ;Interface clear
00121  00FC   TCASY  EQU   0FCH    ;take-control asynchronously
00122  00FD   TCSY   EQU   0FDH    ;Take control synchronously
00123  00FA   TCNTR  EQU   0FAH    ;Take control (receive control)
00124         ;
00125         ;      8292 UTILITY COMMANDS
00126         ;
00127  00E1   WTOUT  EQU   0E1H    ;write to time out register
00128  00E4   RERF   EQU   0E4H    ;read error flag register
00129  00E6   RCST   EQU   0E6H    ;read Controller Status Register
00130  00E7   RBST   EQU   0E7H    ;read GPIB status pseudo-register
00131  000B   IACK   EQU   0BH     ;Interrupt acknowledge
00132         ;
00133         ;      8292   INTERRUPT PORT
00134         ;
00135  0008   PRTF   EQU   08H
00136  0001   TCIF   EQU   01H     ;Task complete interrupt
00137         ;
00138         ;      GPIB  MESSAGES (COMMANDS)
00139         ;
00140  0001   MDA    EQU   1       ;My device address is 1
00141  0041   MTA    EQU   MDA+40H ;My talk address is 1 ("A")
00142  0021   MLA    EQU   MDA+20H ;My listen address is 1 ("!")
00143  003F   UNL    EQU   3FH     ;Universal unlisten
00144  005F   UNT    EQU   5FH     ;Universal untalk
00145  0018   SPE    EQU   18H     ;Serial poll enable
00146  0019   SPD    EQU   19H     ;Serial poll disable
00147  0009   TCT    EQU   9       ;take control (pass control)
00148         ;
00149         ;
00150         ;      CTL PORTS
00151         ;
00152  0080   CTRL1  EQU   80H     ;CTL 8-bit control input
00153  0000   ASR0   EQU   0       ;address switch 1
00154  0001   ASR1   EQU   1       ;address switch 2
00155  0002   ASR2   EQU   2       ;address switch 3
00156  0003   ASR3   EQU   3       ;address switch 4
00157  0004   ASR4   EQU   4       ;address switch 5
00158  0005   DONE   EQU   5     ;DONE status bit
00159  0006   BUSY   EQU   6     ;BUSY status bit
00160  0080   INTRST EQU   80H     ;8291 interrupt status bit
```

```
                ;
0080            CTRLO  EQU    80H     ;TTL 4-bit control output
0001            DNEWT  EQU    1       ;enable WAIT-ON-DONE
0002            SRVC   EQU    2       ;set service request bit
0004            DNECL  EQU    4       ;DONE clear pulse
0010            DMAWT  EQU    10H     ;enable WAIT-ON-DONE
                ;
                ;      NICP PORT (IN/OUT TO NICOLET)
                ;
0040            NICP   EQU    40H
                ;
                ;      MISCELLANEOUS DEFINITIONS
0800          > TABLE  EQU    COMN    ;address of parameter table
0C00            DATBUF EQU    0C00H
0002            DEVICE EQU    2       ;the device number of the camera is 2
0022            CAML   EQU    DEVICE+32
0042            CAMT   EQU    DEVICE+64
0062            INF    EQU    98      ;this sets the input format (am - ct)
0063            XCOORD EQU    99
0069            VII    EQU    105     ;video input increment
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
        ;       OCTOBER 21,1981
        ;
        ;
        ;       MACRO DEFINITIONS
        ;
                MACRO   WRREG   ;REG,VALUE,[LABEL]
        ;       REG=register to write to
        ;       VALUE=value to write
        ;       LABEL=optional jump to
                LD      A,'2'
                OUT     ('1'),A
        ;       ASET    3
                IF      K='@'
                JR      '3'
                ENDIF
                ENDM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                MACRO   CTLNIC
        ;       WRITES BYTES TO NIC FROM CTL
        CTI'@'  IN      A,(CTRLI)
                BIT     DONE,A
                JR      NZ,CTI'@'
                OUTI
                JR      NZ,CTI'@'
                ENDM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                MACRO   NICCTL
                READS BYTES FROM NIC TO CTL
        NII'@'  IN      A,(CTRLI)
                BIT     BUSY,A
                JR      Z,NII'@'
                INI
                OUT     (C),A
                JR      NZ,NII'@'
                ENDM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                MACRO   NICCTL1 ;READS A SINGLE BYTE FROM NIC INTO D-REG
        NII'@'  IN      A,(CTRLI)
                BIT     BUSY,A
                JR      Z,NII'@'
                IN      A,(NICP)
                LD      D,A
                OUT     (NICP),A
                ENDM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                MACRO   STREG   ;REG
        ;       REG=register to be stored at next location addressed by HL
                LD      (HL),'1'
                INC     HL
                ENDM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                MACRO   STSTAT  ;REG
        ;       REG=status register to be stored at ne t location addr     by HL
```

```
                IN      A,('1')
                LD      (HL),A
                INC     HL
                ENDM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                MACRO   STIND   ;INDC
;               INDC=indirect command for accessing the register
                LD      D,'1'
                CALL    RDIND
                LD      (HL),A
                INC     HL
                ENDM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
                        ;       NOVEMBER 4,1981
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                        ;
                        ;       MAIN CONTROL ROUTINE
                        ;
                        ;PURPOSE -- This is the CTL excutive routine which is slave to NIC
                        ;       in that it performs commands issued by NIC and returns to a
                        ;       wait state (waits for input from NIC) after command completion.
                        ;       If a command can not be completed because of a bus error(always
                        ;       considered fatal), the executive returns to its wait state via
                        ;        an error subroutine which, among other things, sets SRVC to
                        ;       indicate abnormal command termination.
                        ;       Each command is started by NIC by transferring a 20 byte parameter
                        ;       block from NIC to CTL.The first byte in this table is the command
                        ;       number.
                        ;       The control routine has been modified such that the CTL address
                        ;       switches may be used to cause a local test program to be run without
                        ;       NIC being attached. See subroutine TEST for a description of the
                        ;       tests and how they are initiated. If ASHO is set to 1, the normal
                        ;       program is executed.
        0000    >               ORG     CSEG
0000 21FF08                     LD      HL,STKBOT
0003 F9                         LD      SP,HL
0004 CD7001     >       WAITO   CALL    INIT
0007 FB                         EI
0008 ED56                       IM      1
000A DB80                       IN      A,(CTRL1)
000C CB47                       BIT     ASHO,A
000E CC6803     >               CALL    Z,TEST
0011 3E14               WAIT    LD      A,14H   ;set parameters for NIC1 table transfer.
0013 320B08     >               LD      (NDAT),A
0016 210008     >               LD      HL,TABLE        ;Starting address of the table
0019 220D08     >               LD      (DATADD),HL
001C 218000     >               LD      HL,WAITZ
001F 221408     >               LD      (RETADD),HL     ;normal return from NIC1.
0022 CD8D02     >               CALL    COMS    ;transfer table
0025 0013                       BLOCK   38H-SCALAR($-CSEG)      ;to get to start of interrupt section
0038 221C08     >       INTRPT  LD      (TEMP),HL       ;the interrupt routine stores registers starting at TEMP
003B E1                         POP     HL
003C 221E08     >               LD      (TEMP+2),HL     ;store PC
003F ED732008>                  LD      (TEMP+4),SP
0043 212208     >               LD      HL,TEMP+6
                                STREG   A
                                STREG   C
                                STREG   B
                                STREG   E
                                STREG   D
                                STSTAT  INT1    ;8291 registers
                                STSTAT  INT2
                                STSTAT  ADRST
                                STSTAT  ADR01
                                STSTAT  EOSR
                                STSTAT  INTST   ;8282 registers
```

```
                                  STIND   RERF     ;error flag
                                  STIND   RCST     ;controller status
                                  STIND   RBST     ;busstatus
0070 C30000                       JP      CSEG
0080 211100  >          WAITZ     LD      HL,WAIT
0083 221408  >                    LD      (RETADD),HL     ;return address for command calls
0086 21A800  >                    LD      HL,START        ;following code is to find address of call
0089 FB                           EI
008A ED56                         IM      1
008C 3A0008  >                    LD      A,(COMN)
008F 4F                           LD      C,A
0090 3A1608  >                    LD      A,(LCOMM)
0093 B9                           CP      C
0094 FCC000  >                    CALL    M,ERROR
0097 79                           LD      A,C
0098 A7                           AND     A
0099 FCC000  >                    CALL    M,ERROR
009C CCC000  >                    CALL    Z,ERROR
009F 0600                         LD      B,0
00A1 3D                           DEC     A
00A2 4F                           LD      C,A
00A3 87                           ADD     A,A
00A4 81                           ADD     A,C
00A5 4F                           LD      C,A
00A6 09                           ADD     HL,BC
00A7 E9                           JP      (HL)
00A8 CD2403  >          START     CALL    COM1
00AB CDB201  >                    CALL    COM2
00AE CD0000  >                    CALL    COM3
00B1 CD3502  >                    CALL    COM4
00B4 CDBD02  >                    CALL    COM5
00B7 CDA702  >                    CALL    COM6
00BA CDBE02  >                    CALL    COM7
00BD CD0009                       CALL    COM8
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                        ;
                        ;         ERROR ROUTINE
                        ;
                        ;PURPOSE - when a fatal error occurs in performing a control ...
                        ;         a call to ERROR is made. ERROR sets the SRVC bit which will cause
                        ;         NIC MONITOR to jump to NIC ERROR. This in turn should cause an
                        ;         interrupt and subsequent state storage.
                        ;
                        ERROR     WRREG   CTRLO,SRVC      ;set the SRVC bit for abnormal return
00C4 DB21                         IN      A,(INT1)
00C6 57                           LD      D,A
00C7 CD5D01  >                    CALL    CTLNIC1 ;send status to NIC
00CA C34101  >                    JP      RETURN
                        ;
                        ;     JANUARY 18,1981
                        ;
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                        ;
```

```
                               ;       LISLIST ROUTINE
                               ;
                               ;PURPOSE--send out a list of listeners or a single talker
                               ;ARGUMENTS-- reg A = 1 => talker list
                               ;                      0 => listener list
                               ;
                               ;USES register A,B,DE
                               ;
00CD 47                        LISLIST         LD      B,A
00CE A7                                AND     A
00CF 200F                              JR      NZ,LI11
00D1 3A0108      >                     LD      A,(NLIST)
00D4 A7                                AND     A
00D5 2003                              JR      NZ,LI3
00D7 04                                INC     B
00D8 1813                              JR      LI4
00DA 47                        LI3     LD      B,A
00DB 110208      >                     LD      DE,LISTP
00DE 1803                              JR      LI1
00E0 110808      >             LI11    LD      DE,TALKP
00E3 CB10                      LI1     RL      B
00E5 1A                        LI5     LD      A,(DE)
00E6 A7                                AND     A
00E7 2803                              JR      Z,LI2
00E9 CD6701      >                     CALL    WAITO
00EC 13                        LI2     INC     DE
00ED 10F6                      LI4     DJNZ    LI5
00EF C9                                RET
                               ;
                               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                               ;
                               ;       BYTBLK ROUTINE
                               ;
                               ;PURPOSE-- sets up registers for block read or write.
                               ;       registers set are B,E,HL
                               ;       register C should be set by the caller for the desired port
                               ;RETURNS--the Z-flag is set if NDAT and NDATB =0
                               ;
00F0 2A0D08      >             BYTBLK  LD      HL,(DATADD)
00F3 3A0B08      >                     LD      A,(NDAT)
00F6 57                                LD      D,A
00F7 1E01                              LD      E,1
00F9 A7                                AND     A
00FA 2805                              JR      Z,BY1
00FC 47                                LD      B,A
00FD 3E01                              LD      A,1
00FF 1806                              JR      BY2
0101 3A0C08      >             BY1     LD      A,(NDATB)
0104 57                                LD      D,A
0105 0600                              LD      B,0
0107 5F                        BY2     LD      E,A
0108 7A                                LD      A,D
0109 A2                                AND     D
```

```
010A C9                 RET
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;       SUBROUTINE RDIND
                ; PURPOSE -- read 8292 indirect registers
                ; ARGUMENTS -- reg D should contain the utility command-
                ; REVC,REERF,RINM,RCST,RBST,RTOUT or RERM
                ;
010B DB11       RDIND   IN      A,(INTST)
010D E602               AND     IBFBT
010F 20FA               JR      NZ,RDIND
0111 7A                 LD      A,D
0112 CD4E01  >          CALL    WAITX
0115 DB10               IN      A,(PRT92)
0117 C9                 RET
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;       SUBROUTINE WRIND
                ; PURPOSE -- write 8292 indirect registers or to send IACK
                ; ARGUMENTS -- reg D should contain WTOUT,WEVC or IACK
                ;               reg E should contain a value to be inserted
                ;               in the indirect reg (except for IACK)
0118 DB11       WRIND   IN      A,(INTST)
011A E602               AND     IBFBT
011C 20FA               JR      NZ,WRIND
011E 7A                 LD      A,D
011F D311               OUT     (CMD92),A
0121 DB11       WR1     IN      A,(INTST)
0123 E602               AND     IBFBT
0125 20FA               JR      NZ,WR1
0127 CB5A               BIT     3,D
0129 2009               JR      NZ,WR2  ; if IACK this is all
012B 7B                 LD      A,E
012C D310               OUT     (PRT92),A
012E DB11       WR3     IN      A,(INTST)
0130 E602               AND     IBFBT
0132 20FA               JR      NZ,WR3
0134 C9         WR2     RET
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;       SUBROUTINE T2IN
                ;PURPOSE -- check GPIB input status
                ; ARGUMENTS -- reg A returns the INT1 status bits.
                ;
0135 1600       T2IN    LD      D,0
0137 DB21       T21     IN      A,(INT1)
0139 B2                 OR      D       ;collect status bits here
013A 57                 LD      D,A
013B CB47               BIT     BIM,A
```

```
0130 28F8            JR     Z,T21
013F 7A              LD     A,D
0140 C9              RET
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                     ;
                     ;      PSEUDO-SUBROUTINE RETURN
                     ;      Returns subroutines to RETADD and writes a byte to NIC
0141 2A1408   >  RETURN LD      HL,(RETADD)
0144 DB80            IN     A,(CTRL1)       ;In test mode a normal return is made.
0146 CB47            BIT    ASR0,A
0148 C8              RET    Z
0149 C1              POP    BC
014A CD5D01   >      CALL   CTLNIC1
014D E9              JP     (HL)
                     ;
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                     ;      SUBROUTINE WAITX
                     ; PURPOSE -- issue 8292 direct command
014E D311        WAITX  OUT    (CMD92),A
0150 DB08        WX2    IN     A,(PRTF)
0152 E601            AND    TCIF
0154 20FA            JR     NZ,WX2
0156 DB08        WX1    IN     A,(PRTF)
0158 E601            AND    TCIF
015A 28FA            JR     Z,WX1
015C C9              RET
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                     ;      SUBROUTINE      CTLNIC1
                     ;      writes a single byte from CTL to NIC
                     ;      the byte should be in the D register
015D DB80        CTLNIC1        IN      A,(CTRL1)
015F CB6F            BIT    DONE,A
0161 20FA            JR     NZ,CTLNIC1
0163 7A              LD     A,D
0164 D340            OUT    (NICP),A
0166 C9              RET
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                     ;
                     ;      SUBROUTINE WAITO
                     ; PURPOSE -- output data on the GPIB
0167 D320        WAITO  OUT    (DOUT),A
0169 DB21        WAIT1  IN     A,(INT1)
016B CB4F            BIT    BOM,A
016D 28FA            JR     Z,WAIT1
016F C9              RET
                     ;
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
                        ;       OCTOBER 27,1981
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                        ;initializaton routine -
                        ;
                        INIT    WRREG   CMD92,RSET
0174 0600                       LD      B,0     ;wait for abort to go out,etc.
0176 10FE               INIT0   DJNZ    INIT0
                                WRREG   INIMR,INIM      ;set TCI interrupt
                                WRREG   AUXMD,INITIAL   ;initialize the 8291
                        INIT2   WRREG   ADR01,DIDL1
                                WRREG   ADR01,DIDL2
                                WRREG   ADRMD,TON
                        INIT3   WRREG   AUXMD,CLFRT
                                WRREG   INT1,0
                                WRREG   INT2,0
                                WRREG   AUXMD,0 ;release initialization
019C 3EF1                       LD      A,GIDL
019E CD4E01   >                 CALL    WAITX
01A1 3E08                       LD      A,LASTC
01A3 321608   '                 LD      (LCOMM),A       ;initialize some storage
01A6 3E04                       LD      A,MXBLK
01A8 321708   '                 LD      (MAXBLK),A
01AB 3E04                       LD      A,4     ;clear busy and done
01AD D340                       OUT     (NICP),A
01AF D380                       OUT     (CTRLO),A
01B1 C9                         RET
                        ;
                        ;       RECV ROUTINE (a las COM2)
                        ;
                        ;PURPOSE--Transfers data from GPIB to CTC (or to other listeners)
                        ;PARAMETERS -
                        ;       (8) primary address-talker
                        ;       (9) secondary address -talker
                        ;       (10) EOS character. (see below)
                        ;       (11) no. of data bytes to receive
                        ;       (12) no. of 256 blocks to receive. (see below)
                        ;       (14) starting address for data storage.
                        ;
                        ;RETURNS - if (11) and (12) are both 0,RECV returns the number of
                        ;       data in (12) and (11). If do not get EOI at end of transmission
                        ;        or if the amount of data received does not match that expected,
                        ;       an error condition exists.
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
01B2 3EFA               COM2    LD      A,TCNTR
01B4 CD4E01   '                 CALL    WAITX
01B7 3E01                       LD      A,1
01B9 CDCD00                     CALL    LISLIST
01BC 0682                       LD      B,AIRA+HOEND
01BE 3A0A08   '                 LD      A,(EOSC)
01C1 A7                         AND     A
01C2 2804                       JR      Z,COM25
01C4 D327                       OUT     (EOSR),A
01C6 0686                       LD      B,AIRA+HOEND+EOEOS
```

```
                         COM25  WRREG   AUXMD,B
                                WRREG   ADRMD,LON
                                WRREG   AUXMD,0
0103 3EF6                       LD      A,GTSE
0105 CD4E01  >                  CALL    WAITX
0108 0E20                       LD      C,DIN
010A CDF000  >                  CALL    BYTBLK
010D 2006                       JR      NZ,COM27
010F 3A1708  >                  LD      A,(MAXBLK)       ;If (I1) and (I2) are 0, go ... no. of data
01E2 5F                         LD      E,A
01E3 0600                       LD      B,0
01E5 DD210000           COM27   LD      IX,0             ;set data counter to 0
01E9 CD3501  >          COM28   CALL    TEIN
01EC E610                       AND     ENDMK
01EE 200B                       JR      NZ,COM29
01F0 DD23                       INC     IX
01F2 EDA2                       INI
01F4 20F3                       JR      NZ,COM26
01F6 1D                         DEC     E
01F7 20F0                       JR      NZ,COM28
01F9 1818                       JR      COM212
01FB EDA2               COM29   INI
01FD DD23                       INC     IX
01FF DD221A08>                  LD      (COUNT),IX
0203 3A0B08  >                  LD      A,(NDAT)         ;if (I1) and (I2) are 0, output the data count
0206 47                         LD      B,A
0207 3A0C08  >                  LD      A,(NDATB)
020A 80                         ADD     A,B
020B 2006                       JR      NZ,COM212
020D 2A1A08                     LD      HL,(COUNT)
0210 220C08  >                  LD      (NDATB),HL
0213 3EF0               COM212  LD      A,TCSY
0215 CD4E01  >                  CALL    WAITX
                                WRREG   AUXMD,HXRA
                                WRREG   ADRMD,TON
                                WRREG   AUXMD,FNHSK
                                WRREG   AUXMD,0
0228 ....                       LD      A,UNT
022A CD6701  >                  CALL    WAITO
022D 3EF1                       LD      A,GTDL
022F CD4E01  >                  CALL    WAITX
0232 C34101  >                  JP      RETURN
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                        ;
                        ;       POLL ROUTINE  (alias COM4)
                        ;
                        ;PURPOSE -- wait for SRQ and do a serial poll of the single
                        ; device requesting service. This routine has been tailored
                        ; to match some of the ideosyncrasies of the Hamamatsu camera.
                        ;PARAMETERS:
                        ;       (B) primary address-device to be polled
                        ;       (Y) secondary address
                        ;
```

```
                        ;RETURNS -- the status byte is sent to NICP if not test mode.
0235 3EFA         COM4   LD      A,TCNTR
0237 CD4E01  >           CALL    WAITX
023A DB11         COM42  IN      A,(INTST)          ;wait for service request
023C CB6F                BIT     5RQ,A
023E CA3A02  >           JP      Z,COM42
0241 3E5F                LD      A,UNT   ;the following sequence is from Hamamats:
0243 CD6701  >           CALL    WAITO           ;"Instruction Manual M999-04 Diagnostics with
0246 3E18                LD      A,SFE   ;Tektronix 4051".
0248 CD6701  >           CALL    WAITO
024B 3E01                LD      A,1
024D CDCD00  >           CALL    LISLIST
                         WRREG   ADRMD,CON
                         WRREG   AUXMD,0
0258 3EF6                LD      A,GTSB
025A CD4E01  >           CALL    WAITX
025D CD3501  >           CALL    T2IN
0260 DB20                IN      A,(DIN)
0262 57                  LD      D,A
0263 DB80                IN      A,(CTRLI)
0265 CB47                BIT     ASR0,A
0267 C45D01  >           CALL    NZ,CTLNIC1
026A 3EFD         COM41  LD      A,TCSY
026C CD4E01  >           CALL    WAITX
026F 3E80                LD      A,TON
0271 D324                OUT     (ADRMD),A
0273 AF                  XOR     A
0274 D325                OUT     (AUXMD),A
0276 3E5F                LD      A,UNT
0278 CD6701  >           CALL    WAITO
027B 3E19                LD      A,SFD
027D CD6701  >           CALL    WAITO
0280 162B                LD      D,IACK+SRQBT    ;clear the SRQ bit on the d292
0282 CD1801  >           CALL    WRIND
0285 3EF1                LD      A,GIDL
0287 CD4E01  >           CALL    WAITX
028A C34101  >           JP      RETURN
                        ;
                        ;
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                        ;
                        ;
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                        ;        NIC1 ROUTINE  (alias COM5)
                        ;
                        ;PURPOSE --read data from NIC to CTL
                        ;PARAMETERS--
                        ;        (11)  no. of bytes or
                        ;        (12) no. of 256 byte blocks
                        ;        (14) starting address for data storage
                        ;
028D 0E40         COM5   LD      C,NICP
028F CDF000  >           CALL    BYTBLK
```

```
0292 2810              JR     Z,COM53
0294 F3                DI
                COM52  NICCTL
02A1 1D                DEC    E
02A2 20F1              JR     NZ,COM52
02A4 C34101  >  COM53  JP     RETURN
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;      NICO ROUTINE  (alias COM6)
                ;
                ;PURPOSE -- write data from CTL to NIC
                ;
                ;PARAMETERS --
                ;      (11) no. of data bytes or
                ;      (12) no. of 256 byte blocks
                ;      (14)  starting address of the data
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
02A7 0E40       COM6   LD     C,NICP
02A9 CDF000  >         CALL   BYTBLK
02AC 280D              JR     Z,COM63
                COM62  CTLNIC
02B8 1D                DEC    E
02B9 20F3              JR     NZ,COM62
02BB C34101  >  COM63  JP     RETURN
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;      REXMT ROUTINE (alias COM7)
                ;
                ;PURPOSE -- Ttransfer data from GPIB to NIC
                ;PARAMETERS -
                ;      (8) primary address-talker
                ;      (9) secondary address-talker
                ;
                ;This routine receives each datum from the GPIB and sends it to
                ; NIC until EOI is detected. Then SRVC is set before the last
                ; datum is sent to NIC.
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
02BE 3EFA       COM7   LD     A,TCNTR
02C0 CD4E01  >         CALL   WAITX
02C3 3E01              LD     A,1
02C5 CDC000  >         CALL   LISLIST
02C8 3E82              LD     A,AXRA+HUEND
                       WRREG  AUXMD,A
                       WRREG  ACRMD,LON
                       WRREG  AUXMD,0
02D5 3EF6              LD     A,GTSB
02D7 CD4E01  >         CALL   WAITX
02DA 0E20              LD     C,DIN
02DC CD3501  >  COM71  CALL   T2IN
02DF E610              AND    ENDM
02E1 200E              JR     NZ,COM72
02E3 ED50              IN     D,(C)
                       CTLNIC
```

```
02EF 18EB               JR      COM71
                COM72   WRREG   CTRL0,SRVC
                        CTLNIL
02FF AF                 XOR     A
0300 D380               OUT     (CTRL0),A       ;clear SRVC
0302 3EFD               LD      A,TCSY
0304 CD4E01   >         CALL    WAITX
                        WRREG   AUXMD,AXRA
                        WRREG   ADRMD,TON
                        WRREG   AUXMD,FNHSK
                        WRREG   AUXMD,0
0317 3E5F               LD      A,UNT
0319 CD6701   >         CALL    WAITO
031C 3EF1               LD      A,GIDL
031E CD4E01   >         CALL    WAITX
0321 C34101   >         JP      RETURN
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;       OCTOBER 27,1981
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;       SEND ROUTINE (ALIAS COM1)
                ;
                ;       sends data from CTL to the GPIB
                ;
                ;INPUT  (1) no. of listeners
                ;       (2) primary address-first listener
                ;       (3) secondary address - first listener
                ;       (4) --second listener
                ;       (5) --
                ;       (6) --third listener
                ;       (7)--
                ;       (10)  EOS character (if EOS=0 then EOI is sent with the last byte
                ;       according to the data count given by (10) and (11). If EOS is
                ;       non-zero ,the no. of characters sent is determined by the sequential
                ;       location of the EOS character in the data stream provided that
                ;       the amount of data as determined by (10) or (11) is greater or
                ;       or equal to this location. Otherwise EOI is suppressed.
                ;       (11)  no. of bytes to send or
                ;       (12)  no. of 256 byte blocks to send. (If both (11) and (12) are 0
                ;        no data is sent.
                ;       (14)  sarting address of th data to be sent.
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0324 3EFA       COM1    LD      A,TCNTR
0326 CD4E01   >         CALL    WAITX
0329 AF                 XOR     A
032A CD0000   >         CALL    LISLIST
032D 3EF6               LD      A,GTSB
032F CD4E01   >         CALL    WAITX
0332 0E20               LD      C,DOUT
0334 CDF000   >         CALL    BYTBLK
0337 281D               JR      Z,COM17
0339 1601               LD      D,1
033B 78         COM15   LD      A,B
```

```
0330 BA                   CP     D
033D 2008                 JR     NZ,COM18
033F 7B                   LD     A,E
0340 BA                   CP     D
0341 2004                 JR     NZ,COM18        ;if B=0 and E=1, this is the last byte to be sent
                          WRREG  AUXMD,SEOI
0347 EDA3          COM18  OUTI
0349 2805                 JR     Z,COM19
034B CD6901  >            CALL   WAIT1
034E 18EB                 JR     COM15
0350 CD6901  >     COM19  CALL   WAIT1
0353 1D                   DEC    E
0354 20E5                 JR     NZ,COM15
0356 3EFD          COM17  LD     A,TCSY
0358 CD4E01  >            CALL   WAITX
035B 3E3F                 LD     A,UNL
035D CD6701  >            CALL   WAITO
0360 3EF1                 LD     A,GIDL
0362 CD4E01  >            CALL   WAITX
0365 C34101  >     COM110 JP     RETURN
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ;
                   ;
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ;
                   ; TEST ROUTINE
                   ;
                   ; REVISION -- OCTOBER 21,1981
                   ; AUTHOR   -  BARRETT,TB (PML)
                   ; PURPOSE  -- performs a camera test independently of NIC. This routine
                   ;       is initiated if the CIL address bit 0 is set to 0 and at least one other
                   ;       address bit is set to 1. (At present only bits 1 and 2 will cause a test.)
                   ;       The following tests are done:
                   ;       ASR1=1  => initialize cursor to left margin and step to the right
                   ;               margin using the default interlace, then leave the cursor at the
                   ;               default position.
                   ;       ASR2=1  => use the default cursor position and interlace and gather
                   ;               one line of video (binary mode).
                   ;
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0368 3E01          TEST   LD     A,1     ;1 listener
036A 320108  >            LD     (NLIST),A
036D 21000C               LD     HL,DATBOT
0370 220D08  >            LD     (DATADD),HL
0373 214200               LD     HL,CAMT
0376 220808  >            LD     (TALKP),HL
0379 AF                   XOR    A
037A 320A08  >            LD     (EOSC),A
037D DB80                 IN     A,(CTRL1)
037F CB4F                 BIT    ASR1,A
0381 2830                 JR     Z,TEST2
0383 212263               LD     HL,(XCOORD SHL 8)'CAML  ;Camera secondary and primary address for XCOORD
0386 220208  >            LD     (LISTP),HL
```

```
00830 0389 210400          LD      HL,0004H        ;4 data bytes for each coordinate
00831 038C 220B08  >       LD      (NDAT),HL
00832 038F 210001          LD      HL,0100H        ;set first coord. to 0
00833 0392 110004          LD      DE,0400H  ;count to 1024
00834 0395 E5        TEST1 PUSH    HL
00835 0396 D5              PUSH    DE
00836 0397 CDFF03  /       CALL    BCD+BC
00837 039A CD2403  /       CALL    COM.
00838 039D D1              POP     DE
00839 039E E1              POP     HL
00840 039F 3E01            LD      A,1
00841 03A1 44              LD      B,H
00842 03A2 80              ADD     A,B
00843 03A3 27              DAA
00844 03A4 67              LD      H,A
00845 03A5 3E00            LD      A,0
00846 03A7 45              LD      B,L
00847 03A8 88              ADC     A,B
00848 03A9 27              DAA
00849 03AA 6F              LD      L,A
00850 03AB 1D              DEC     E
00851 03AC 20E7            JR      NZ,TEST1
00852 03AE 15              DEC     D
00853 03AF 20E4            JR      NZ,TEST1
00854 03B1 DB80            IN      A,(CTRL1)
00855 03B3 CB57      TEST2 BIT     ASR,A
00856 03B5 2844            JR      Z,TEST3
00857 03B7 212262          LD      HL,(INF SHL 8)+CAML     ;set input format to binary
00858 03BA 220208  >       LD      (LIST1P),HL
00859 03BD 210100          LD      HL,0001H        ;send one byte
00860 03C0 220B08  >       LD      (NDAT),HL
00861 03C3 3E02            LD      A,2     ;code for binary output
00862 03C5 32000C          LD      (DATBOT),A
00863 03C8 CD2403  >       CALL    COM1
00864 03CB 010200          LD      BC,0002H
00865 03CE C5        TEST21 PUSH   BC
00866 03CF 210000          LD      HL,0000H        ;no data is sent with command VII
00867 03D2 220B08  >       LD      (NDAT),HL
00868 03D5 3E09            LD      A,VII
00869 03D7 32030B  >       LD      (LIST5),A
00870 03DA CD2403  /       CALL    COM1
00871 03DD 1600            LD      D,0     ;pause for camera data gather
00872 03DF 42              LD      B,D
00873 03E0 10FE      Test20 DJNZ   TEST20
00874 03E2 15              DEC     D
00875 03E3 20FB            JR      NZ,TEST20
00876 03E5 AF              XOR     A
00877 03E6 32030B  >       LD      (LIST5),A
00878 03E9 CD3502  >       CALL    COM4
00879 03EC 210002          LD      HL,0200H        ;2 blocks, 0 bytes
00880 03EF 220B08  >       LD      (NDAT),HL
00881 03F2 CD8201  >       CALL    COM2
00882 03F5 C1              POP     BC
```

```
03F6 10D6             DJNZ   TEST21
03F8 0D               DEC    C
03F9 20D3             JR     NZ,TEST21
03FB CD7001  >  TEST3 CALL   INIT   ;reset to default values
03FE C9               RET
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;     SUBROUTINE BCDASC
                ;
                ; given a two byte packed BCD number at HL with lowest order
                ; nibbles in H, BCDASC places the equivalent ASCII code at
                ; DATBOT starting with the most significant nibble.
                ;
03FF DD21000C   BCDASC LD    IX,DATBOT
0403 220F08  '         LD    (DUMIL),HL
0406 210F08  >         LD    HL,DUMIL
0409 0604              LD    B,4
040B 3E30              LD    A,30H
040D ED6F       BCD1   RLD
040F DD7700            LD    (IX+0),A
0412 DD23              INC   IX
0414 05                DEC   B
0415 C8                RET   Z
0416 CB40       BCD2   BIT   0,B
0418 20F3              JR    NZ,BCD1
041A 23                INC   HL
041B 18F0              JR    BCD1
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                          ;
                          ;       PARAMETER TABLE
                          ;
            0800    )             ORG     DSEG
      0800 00             COMM    BYTE    0       ;command number
      0801 00             NLIST   BYTE    0       ;no. of listeners (or can be set to 1 talker)
      0802 00             LIS1P   BYTE    0       ;primary address,1st listener (or a talker)
      0803 00             LIS1S   BYTE    0       ;secondary address,first listener (or 1 talker)
      0804 00             LIS2P   BYTE    0       ;--second listener
      0805 00             LIS2S   BYTE    0
      0806 00             LIS3P   BYTE    0       ;--third listener
      0807 00             LIS3S   BYTE    0
      0808 00             TALKP   BYTE    0       ;talker primary address
      0809 00             TALKS   BYTE    0       ;talker secondary address
      080A 00             EOSC    BYTE    0       ;EOS character (0 means none exists)
      080B 00             NDAT    BYTE    0       ;no. of data bytes to be transferred, or
      080C 00             NDATB   BYTE    0       ;no. of 256 byte blocks to be transferred
      080D 0000           DATADD  WORD    0       ;starting address of the data block in CTL RAM
      080F 00             DUMIL   BYTE    0
      0810 00             DUMIH   BYTE    0
      0811 00             MESS    BYTE    0       ;message print indicator
      0812 00             STAT1   BYTE    0
      0813 00             STAT2   BYTE    0
                          ;
                          ;       OTHER DATA
                          ;
      0814 0000           RETADD  WORD    0
      0816 08             LCOMM   BYTE    8       ;set to the last valid command number
      0817 04             MAXBLK  BYTE    4       ;set to the max. no. of 256 byte blocks for data
      0818 0000           STACKP  WORD    0       ;temporary storage for stack-pointer
      081A 0000           COUNT   WORD    0       ;data counter location
                          ;
                          ;       TEMPORARY STORAGE
                          ;
      081C 00             TEMP    BYTE    0
      081D 00             T2      BYTE    0
```

Strings and Macros

CTLNIC - 007A M	NICCTL - 0080 M	NICCTL1 - 0097 M	STIND - 009F M	STREG - 00F5 M
STSTAT - 004C M	WRREG - 007F M			

Vars

ABORT -- 00F9	ADRV1 -- 0026	ADRMD -- 0024	ADRST -- 0024	ASK0 --- 0001
ASK1 --- 0001	ASK2 --- 0002	ASK3 --- 0003	ASK4 --- 0004	AUXMD -- 0025
AWAIT --- 0080	AXRB --- 00AC	BIM ---- 0000	BOM ---- 0001	BUSST -- 0010
BUSY --- 0006	CABT --- 004D	CAML --- 0022	CAMT --- 0042	CLKRT -- 0024
CLRST -- 0010	CMD92 -- 0011	COM8 --- 0900	CPT ---- 0180	CPTEN -- 0001
CRC16 -- 0025	CSEB --- 0000	CTRLI -- 0080	CTRLO -- 0080	DATBOI -- 0C00
DEVICE - 0002	DIN ---- 0020	DMAMT -- 0010	DNECL -- 0004	DNEMT -- 0001
DOPL --- 0005	DOUT --- 0020	DSEG --- 0800	DTDL1 -- 0060	DTDL2 -- 00E0
EDEOS -- 0004	ENDMA -- 0010	EOIS --- 0008	EOSR --- 0027	ENFLAG - 0010
EOT --- 0040	ERRM --- 0010	FNMSK -- 0003	GIDL --- 00F1	GTSB --- 00F8
HDERD -- 0002	IACK --- 0008	IBFBT -- 0002	INF ---- 0002	INITIAL - 0002
INT1 --- 0021	INT2 --- 0022	INTBST - 0080	INTM --- 0004	INTMK --- 0020
INTST -- 0011	K ------ 0003 V	LASTC -- 0008	LON ---- 0040	MDA ---- 0001
MLA ---- 0021	MODE1 -- 0001	MTA ---- 0041	MXBLK -- 0004	NICP --- 0140
PRT91 -- 0020	PRT92 -- 0010	PRTF --- 0008	RBST --- 00E7	RCST --- 00E8
RERF --- 00E4	RSET --- 00F2	RSTI --- 00F3	SEQI --- 0006	SFD ---- 0019
SPE ---- 0018	SRQ ---- 0045	SRQBT -- 0020	SRVC --- 0002	STRBOI - 0FFF
SYCBT -- 0008	SYCS --- 0008	TA ----- 0002	TCASY -- 0001	TCLD --- 0001
TCNTR -- 00FA	TCSY --- 00F0	TCT ---- 0009	TMOUT -- 0002	TON ---- 0080
TDXEG -- 0010	TDSI --- 0010	TOUT2 -- 0002	TOUT3 -- 0004	UNL ---- 003F
UP ----- 005F	VII ---- 0069	VSCMD -- 000F	WTOUT -- 00E1	XCJUMP - 0006

INIT (default) Section (081E)

BCD1 --- 040D	BLD2 --- 0416	BCDASC - 03FF	BY1 ---- 0101	BY2 ---- 0102
BYTBLK - 00F0	COM1 --- 0324 G	COM110 - 0365	COM15 -- 0336	COM17 -- 0350
COM19 -- 0347	COM19 -- 0350	COM2 --- 0182 G	COM212 - 0213	COM.5 --- 01C8
COM2 -- 01E5	COM28 -- 01E9	COM29 -- 01FB	COM4 --- 0235 G	COM41 --- 020A
COM42 -- 023A	COM5 --- 0280 G	COM52 -- 0295	COM53 -- 02A4	COM6 --- 02A7 G
COM62 -- 02AE	COM63 -- 02B8	COM7 --- 02BE	COM71 -- 02CC	COM72 --- 02F1
COMN --- 0800	COUNT -- 081A	CT14200 02AE	CT18200 02E5	CTM2000 02F5
DATACI 015D	DATADD - 0800	DUM1M -- 0810	DUM1L -- 0804	EOSC --- 0804
ERR ---- 00C9	INIT --- 0170	INIT0 -- 0176	INIT2 -- 0180	INIT3 --- 018C
INPUT - 0038	LCOMM -- 0816	LI1 ---- 00E3	LI11 --- 00E9	LI2 ---- 00EF
LI3 ---- 00DA	LI4 ---- 00ED	LI5 ---- 00E5	LI51P -- 0301	LI51S -- 0307
LI52P -- 0804	LI52S -- 0805	LI53P -- 0806	LI53S -- 0807	LI5L151 00EB
MAXBLK - 0817	MESS --- 0811	NDAT --- 0808	NDATB -- 080C	NII2000 0249
NLIST -- 0801	RDIND -- 0108	RETADD - 0814	RETURN - 0141	STACKP - 081A
SP ----- 00A8	STAT1 -- 0812	STAT2 -- 0813	T2 ----- 0010	T21 ---- 0137
TZIN --- 0135	TABLE -- 0800	TALKP -- 0808	TALKS -- 0809	TEMP --- 0811
TEST --- 0368	TEST1 -- 0395	TEST2 -- 03B3	TEST20 - 03E0	TEST21 - 03EE
TOP ---- 03FB	WAIT --- 0011	WAIT1 -- 0169	WAIT0 -- 0167	WAIT0 -- 0004
XR ----- 014E	WAIT2 - 0080	WR1 ---- 0121	WR2 ---- 0134	WR3 ---- 012E
WR 4 -- 011B	WX1 ---- 0156	WX2 ---- 0150		

APPENDIX G - TEKREC Source Listing

```
PL/I-80 V1.3  COMPILATION OF: TEKREC

   1   0000 /* TEKREC
   2   0000  REVISION -- October 20,1981
   3   0000  AUTHOR   -- Barrett,IB  (PML)
   4   0000  PURPOSE  -- Store a Tektronix Hexadecimal (Tekhex) file as a
   5   0000  CP/M COM file. The error-free reception of each record is
   6   0000  acknowledged by ACK followed by CR and a prompt character of LF.
   7   0000 Thus Tek COMM should include the parameter P=0A.
   8   0000 Reception of an abort block will cause an error code 1.
   9   0000
  10   0000 */
  11   0000
  12   0000
  13 a 0000 tekrec:
  14 a 0006     proc    options(main);
  15 c 0006     %replace
  16 c 0006         cret by '0D',
  17 c 0006         lf by '0A',
  18 c 0006         ack by '06',
  19 c 0006         nak by '15',
  20 c 0006         true by '1'b,
  21 c 0006         false by '0'b;
  22 c 0006     dcl
  23 c 0006         (punch,dis,reader,sysprint)    file,
  24 c 0006         checksum    fixed(7),
  25 c 0006         (line,col,nbytes)    fixed(15),
  26 c 0006         (check,type)    bit(8),
  27 c 0006         start       bit(16),
  28 c 0006         length      bit(8) static init('1E'b4),
  29 c 0006         length1     bit(8) static init('1E'b4),
  30 c 0006         (p,pdata,ptype,plength,pcheck,pbdata)
  31 c 0006                     pointer,
  32 c 0006         1 breakdown based(p),
  33 c 0006           2 lostart   bit(8),
  34 c 0006           2 histart   bit(8),
  35 c 0006         rstart          fixed(15) based(p),
  36 c 0006         fdata(425,30)       fixed(7) based(pdata),
  37 c 0006         ftype           fixed(7) based(ptype),
  38 c 0006         flength         fixed(7) based(plength),
  39 c 0006         fcheck          fixed(7) based(pcheck),
  40   0001         name            char(14) varying,
  41   0003         (rec_data,send_data,chr)     char(1),
  42 c 0006         back_tab            char(4),
  43 c 0006         data(425,30) bit(8),
  44 c 0006         bdata(128) bit(8) based(pbdata),
  45 c 0006         adata(100,128) bit(8) based(pdata);
  46 c 0006
  47   0006     on error
  48 d 0014         begin;
  49 e 0017         dcl code fixed(15);
  50 e 0017         code=oncode();
  51 e 001D         put skip edit('Error',code) (a,x(1),f(8));
  52 e 004A         if code = 64 then stop;
  53 e 0059         put skip edit('length=',length) (a,b4(2));
  54 e 0086         send_data=nak;
  55 e 0092         get edit(chr) (a);
  56 e 00AF         put skip(2) edit('Reading Record # ') (a);
  57   0101         go to cont;
```

```
 58 d 00D4          end;
 59 c 00D4      back_tab = ascii(8)||ascii(8)||ascii(8)||ascii(8);
 60 c 0100      p = addr(start);
 61 c 0106      pdata = addr(data(1,1));
 62 c 010C      ptype = addr(type);
 63 c 0112      plength = addr(length);
 64 c 0118      pcheck = addr(check);
 65 c 011E      open file(sysprint) title('$con') stream output linesize(0);
 66 c 013A      put skip edit('TEMREC:  RECEIVE A FILE AT THE TTY PORT')(a);
 67 c 015C      open file(reader) title('$rdr') stream input linesize(0);
 68 c 0178      open file(punch) title('$pun') stream output linesize(0);
 69 c 0194      put skip edit('Enter filename for destination system:  ') (a);
 70 c 01B6      get edit(name) (a);
 71 c 01D3      open file(disk) title(name) output sequential env(b(128));
 72 c 01EE      rec_data = ack;
 73 c 01FA      put skip(2) edit('Destination file:',name)(a);
 74 c 0225      put skip(2) edit('Reading Record 0   ')(a);
 75 c 0247      line=1;
 76 c 024D      nbytes=0;
 77 c 0251      do while(length ='00'b4);
 78 c 0259          call send(ack);
 79 c 025F          rec_data=ack;
 80 c 026B              do while (rec_data ^= '/');
 81 c 027A              get file(reader) edit(rec_data) (a(1));
 82 c 029D              end;
 83 c 029D          get file(reader) edit(start,length,type)
 84 c 02CF              (b4(4),b4(2),b4(2));
 85 c 02CF              if length='00'b4 then go to cont;
 86 c 02DA              if lostart='2F'b4 then signal error(64);
 87 c 02EA              get file(reader) edit((data(line,col) do col=1 to flength),check)
 88 c 0350                  (3ob4(2),b4(2));
 89 c 0350              checksum=0;
 90 c 0354                  do col=1 to flength;
 91 c 0372                  type=data(line,col);
 92 c 038B                  checksum=checksum+substr(type,1,4)+substr(type,5,4);
 93 c 03b9                  end;
 94 c 03B9              if checksum=fcheck then
 95 c 03C3                  do;
 96 c 03C3                  sen data=ack;
 97 c 03CF                  line=line+1;
 98 c 03D6                  nbytes=nbytes+flength;
 99 c 03EA                  end;
100 c 03EA              else
101 c 03EA                  sen data=nak;
102 c 03F6              put edit(back_tab,start) (a(4),b4(4));
103 c 0425      cont:   call send(sen_data);
104 c 042E      end;
105 c 042E      line=nbytes/128;
106 c 043C      if line*128 ^=nbytes then line=line+1;
107 c 0459          do col=1 to line;
108 c 0472          pbdata=addr(adata(col));
109 c 0486          write file(disk) from(bdata);
110 c 04A8          end;
111 c 04A8      put skip edit('nbytes=',nbytes,',n-recs=',line) (2(a,f(6)));
112 c 04E8      close file(disk);
113 c 04F1 send:
114 c 04F1      proc    (data);
115 e 04F1      dcl
116 e 04FB          data    char(1);
117 e 04FB      put file(punch) edit(data)(a(1));
```

```
118 c 051C end send;
119 c 051C
120 a 051C end texrec;
```

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, solid state sciences, electromagnetics and electronic reliability, maintainability and compatibility.

Printed by
United States Air Force
Hanscom AFB, Mass. 01731

END

FILMED

3-85

DTIC